hexagon:2611.00374v1

EXACT SIZES OF NETWORKS THAT SELECT THE THREE OR FOUR LARGEST OF n ELEMENTS

TEJASVI SINGH TOMAR

Abstract. We study (t, n)-selection networks, which move the t largest of n inputs to t given lines in any order, and the least number $\hat{U}_t(n)$ of comparators in such a network. Alekseev showed that $\hat{U}_t(n) \geq (n-t)\lceil \lg(t+1) \rceil$. For $t = 2^k \geq 4$ we raise this bound by $k-1$ for every $n > t$, and by $2k-5$ when $t \geq 32$; as far as we know, these are the first improvements at powers of two. The proofs follow the weights of the lines, and a pruning step of Yao extends the gain to every $t \geq 3$. For $t = 3$, Yao's upper bound $2n - 5 + \lfloor \lg(n-3) \rfloor$ is exact for $4 \leq n \leq 18$. For $t = 4$ the new bound is $3n - 11$ for all $n \geq 5$, and the value $3n - 10$ that Knuth expected is exact for $6 \leq n \leq 12$. Their lower bounds rest on exhaustive searches at a few sizes and on the fact that one more input costs at least two more comparators. Most of the results are also proved in Lean 4, some through compiled checks.

1. Introduction

Selection networks, comparator networks that move the t largest of n inputs to t given lines, appear in Knuth's treatment of sorting networks [8, Sec. 5.3.4] and, more recently, as encodings of cardinality constraints in SAT and constraint solving [3, 2, 6]. Their least size is known only in a few cases.

A comparator network on the lines $0, \ldots, n-1$ is a sequence of comparators $[i{:}j]$ with $i < j$; each puts the smaller of its two values on line i, its *lower* line, and the larger on line j, its *upper* line. The network *selects the t largest*, or is a *(t, n)-selector*, if for every input the t largest values end on the lines $n-t, \ldots, n-1$, in some order. Following Knuth [8, Sec. 5.3.4], $\hat{U}_t(n)$ denotes the least number of comparators in a (t, n)-selector. Clearly $\hat{U}_1(n) = n-1$. Mirroring the lines and exchanging minima and maxima turns a (t, n)-selector into an $(n-t, n)$-selector of the same size, so $\hat{U}_t(n) = \hat{U}_{n-t}(n)$.

Alekseev [1] proved $\hat{U}_t(n) \geq (n-t)\lceil \lg(t+1) \rceil$, where $\lg = \log_2$. The bound is exact for $t = 2$, where $\hat{U}_2(n) = 2n - 4$ [8, Sec. 5.3.4]. When t is not a power of two, Yao [16, Thm. 2.5] proved $\hat{U}_t(n) \geq \lceil \lg(t+1) \rceil\, n + (t - 2^{\lfloor \lg t \rfloor}) \lg n + C(t)$ for some function $C(t)$, so Alekseev's bound is not exact for large n. When $t \geq 4$ is a power of two, Alekseev's bound is the best lower bound for general n that we found in the literature. For $t = 3$, Yao [16, Thms. 2.6 and 2.7] (see also [14, 15]) proved

$$2n - 6 + \lceil \lg \lceil n/3 \rceil \rceil \;\leq\; \hat{U}_3(n) \;\leq\; 2n - 5 + \lfloor \lg(n-3) \rfloor, \tag{1}$$

so that $\hat{U}_3(n) = 2n + \lg n + O(1)$, and he listed the exact value as open [16, Sec. 2.5]. Apart from the values $\hat{U}_3(5) = 6$ and $\hat{U}_3(6) = 8$, from which the recursion of his upper bound starts, we found no exact value of $\hat{U}_3(n)$ with $n \geq 5$ in the literature. For $t = 4$, Knuth's exercise 5.3.4–20, in its 2003 form [7], asks to prove $\hat{U}_4(n) \leq 3n - 10$ for $n \geq 6$. Its answer credits "a subtle construction

Date: 30 October 2026.

2020 *Mathematics Subject Classification.* Primary 68P10; Secondary 68Q06.

Key words and phrases. Selection networks, comparator networks, lower bounds, exhaustive search, formal verification.

by M. W. Green" and adds "(Equality probably holds.)" Qu, Tasbihi and Kschischang [10] show a $(4, 8)$-selector with 14 comparators and call it conjectured-optimal. Constructions for large n go back to Alekseev and Yao; see also Pippenger [9], Jimbo and Maruoka [5] and the encodings cited above. None of these works determines $\hat{U}_t(n)$ outside the cases above.

Results.

Theorem A. *Let $t = 2^k$ with $k \geq 2$. Then $\hat{U}_t(n) \geq (k+1)(n-t) + k - 1$ for every $n \geq t+1$. In particular $\hat{U}_4(n) \geq 3n - 11$ for $n \geq 5$, $\hat{U}_8(n) \geq 4n - 30$ for $n \geq 9$, and $\hat{U}_{16}(n) \geq 5n - 77$ for $n \geq 17$.*

Theorem B. *Let $t = 2^k$ with $k \geq 4$. Then $\hat{U}_t(n) \geq (k+1)(n-t) + 2k - 5$ for every $n \geq t+1$. In particular $\hat{U}_{32}(n) \geq 6n - 187$ for $n \geq 33$ and $\hat{U}_{64}(n) \geq 7n - 441$ for $n \geq 65$.*

Theorem C. *Let $t = 2^k$ with $k \geq 2$. Then $\hat{U}_t(n) \geq (k+1)(n-t) + k - 1 + \mathrm{LB}(k)$ for every $n \geq t+1$, where $\mathrm{LB}(k) \geq \max(0, k-4)$ is the minimum (4) of Section 5, taken over the possible final weights of the output lines.*

Corollary D. *Let $t \geq 2$, $k = \lfloor \lg t \rfloor$ and $n \geq t+1$, and let $\delta_k = \max(0, k-4)$. Then*

$$\begin{aligned}\hat{U}_t(n) &\geq (k+1)(n-t) + k - 1 + \delta_k + \sum_{i=0}^{t-2^k-1} \lceil \lg \lceil \tfrac{n-i}{t-i} \rceil \rceil \\ &\geq (n-t)\lceil \lg(t+1) \rceil + t - 2^k + k - 1 + \delta_k.\end{aligned}$$

Since $\lceil \lg(t+1) \rceil = k+1$ when $t = 2^k$, Theorem A raises Alekseev's bound by $k - 1 = \lg t - 1$ at every power of two $t \geq 4$, and Theorem B by $2k - 5$ when $t \geq 32$. Theorem C contains both, as $\mathrm{LB}(k) \geq \max(0, k-4)$, and it is stronger for $9 \leq k \leq 20$ (Remark 5.4, computed). The three proofs use only the weights of the lines, and for $t = 4$, 8, 16, 32 and 64 and $n \geq t+3$ their bounds are the best that the weights can give (Proposition 5.5). Corollary D follows from Theorems A and B by a pruning step of Yao. For $t = 3$ the first bound of the corollary is Yao's lower bound in (1). For fixed t that is not a power of two, Yao's bound exceeds Alekseev's by $(t - 2^k) \lg n + O(1)$, and so does the first bound of the corollary; as each term of its sum is at least $\lg(n/t)$, it gives Yao's Theorem 2.5 for $n > t$ with $C(t) = k - 1 + \delta_k - (k+1)t - (t - 2^k) \lg t$. The second bound is weaker for large n, but it adds $t - 2^k + k - 1 + \delta_k$ to Alekseev's bound for every $n > t$. This is at least 1 for $t \geq 3$, so for $n > t$ Alekseev's bound is exact only when $t \leq 2$. Both bounds are attained at $n = t+1$ for $2 \leq t \leq 7$, where $\hat{U}_t(t+1) = \hat{U}_1(t+1) = t$. For $t = 4$, Theorem A and the bound of Knuth's exercise show that $\hat{U}_4(n)$ is $3n - 11$ or $3n - 10$ for every $n \geq 6$ (Corollary 5.7), and Theorem 1.3 below decides between them for $n \leq 12$.

For $t = 3$ and $t = 4$ we determine $\hat{U}_t(n)$ for small n.

Theorem 1.1. *$\hat{U}_3(n) = 2n - 5 + \lfloor \lg(n-3) \rfloor$ for $4 \leq n \leq 18$. In particular $\hat{U}_3(n) = 2n - 2$ for $11 \leq n \leq 18$.*

Theorem 1.2. *$\hat{U}_3(n) \geq 2n - 2$ for every $n \geq 11$.*

The bound $2n - 2$ of Theorem 1.2 minus the lower bound in (1) is $4 - \lceil \lg \lceil n/3 \rceil \rceil$: 2 for $n = 11, 12$, 1 for $13 \leq n \leq 24$, 0 for $25 \leq n \leq 48$, and negative from $n = 49$ on. Beyond Theorem 1.1, it is new only for $19 \leq n \leq 24$, where it leaves $\hat{U}_3(n) \in \{2n - 2, 2n - 1\}$ (Table 1).

Theorem 1.3. *$\hat{U}_4(n) = 3n - 10$ for $6 \leq n \leq 12$.*

For $n = 8$ this settles the conjecture of Qu, Tasbihi and Kschischang, and for $7 \leq n \leq 12$ it confirms Knuth's expectation. The case $n = 6$ was known, since $\hat{U}_4(6) = \hat{U}_2(6) = 8$, and

$\hat{U}_4(7) = \hat{U}_3(7) = 11$ follows from Theorem 1.1. For $n = 11$ and $n = 12$ the step from the bound $3n - 11$ of Theorem A to $3n - 10$ rests on the searches of Section 6 alone.

Lemma 1.4 (step lemma)**.** *If $t \geq 2$ and $n \geq t + 1$, then $\hat{U}_t(n) \geq \hat{U}_t(n-1) + 2$.*

The lemma is the analogue for $\hat{U}_t$ of Knuth's inequality $\hat{V}_3(n) \geq \hat{V}_3(n-1) + 2$ for $n \geq 4$, where $\hat{V}_3(n)$ is the least size of a network that moves the third largest of n inputs to a given line [7, answer to exercise 5.3.4–20(a)]. We prove the lemma in Section 4 by a different argument. Applied repeatedly from $n = t$, it yields Alekseev's bound $2(n - t)$ for $t = 2, 3$.

n	Alekseev $2(n-3)$	Yao's lower bound	$\hat{U}_3(n)$	Yao's upper bound
4	2	3	3	3
5	4	5	6	6
6	6	7	8	8
7	8	10	11	11
8	10	12	13	13
9	12	14	15	15
10	14	16	17	17
11	16	18	20	20
12	18	20	22	22
13	20	23	24	24
14	22	25	26	26
15	24	27	28	28
16	26	29	30	30
17	28	31	32	32
18	30	33	34	34
19	32	35	≥ 36	37
20	34	37	≥ 38	39
21	36	39	≥ 40	41
22	38	41	≥ 42	43
23	40	43	≥ 44	45
24	42	45	≥ 46	47

TABLE 1. The values of $\hat{U}_3(n)$ for $n \leq 18$ (Theorem 1.1) and the lower bound of Theorem 1.2 for $19 \leq n \leq 24$, with the bounds in (1).

For $t = 3$ the upper bounds come from Yao's construction [16, Thm. 2.7]. For $t = 4$, the bound $\hat{U}_4(n) \leq 3n - 10$ is Knuth's exercise [7, answer to exercise 5.3.4–20(b)]. Our upper bounds for $8 \leq n \leq 12$ are the optimal networks that our searches output, checked on all inputs (Section 3); those for $n = 6, 7$ come from $\hat{U}_4(n) = \hat{U}_{n-4}(n)$. The lower bounds for $t = 3$ need only two exhaustive searches, at $n = 7$ and $n = 11$. These are the sizes $n = 2^j + 3$ with $5 < n \leq 18$, at which Yao's upper bound grows by three; at $n = 5$ no search is needed, since $\hat{U}_3(5) = \hat{U}_2(5)$ is known. Lemma 1.4 yields the rest. For $t = 4$, by contrast, each n needs its own search, which raises the bound $3n - 11$ of Theorem A to $3n - 10$.

Section 2 reduces selection to a finite problem on the inputs of weight t, Section 3 gives the upper bounds, and Section 4 proves Lemma 1.4. Section 5 proves Theorems A–C by following the weights of the lines in time order, bounds what this method can give (Proposition 5.5), and derives Corollary D with a pruning step of Yao that mirrors the proof of Lemma 1.4. Section 6 describes the searches and their certificates, and Section 7 proves Theorems 1.1–1.3. Section 8 reduces Yao's question to the sizes $2^j + 3$ and gives an exact criterion for each of them. Section 9

tests lead-pattern potentials for $t = 3$ on the inputs of weights 2 and 3, and Section 10 describes the formal proofs.

2. Preliminaries

The comparators $[i{:}j]$ of Section 1, with $i < j$, are *standard*. A *generalized* comparator (a, b), with $a \neq b$, puts the smaller value on line a and the larger on line b. A network of generalized comparators can be untangled into a standard network of the same size whose output is the original output with the lines permuted; this is well known (see [8, Sec. 5.3.4]). Comparator networks commute with monotone maps of the values, so they are determined by their action on 0-1 inputs, which they map to 0-1 vectors of the same weight. For $0 \leq t \leq n$ let $V_{n,t} \subseteq \{0,1\}^n$ be the set of the $\binom{n}{t}$ vectors of weight t, and write $\mathbf{1}_{\text{top}}$ for the vector with ones exactly on the lines $n-t, \ldots, n-1$.

Lemma 2.1. *A standard network is a (t, n)-selector if and only if it maps every vector of $V_{n,t}$ to $\mathbf{1}_{\text{top}}$.*

Proof. The condition is necessary. Conversely, let x be a 0-1 input of weight w. If $w \leq t$, choose $y \geq x$ in $V_{n,t}$; by monotonicity the output of x is at most $\mathbf{1}_{\text{top}}$ in every coordinate, so its ones are on the top t lines. If $w \geq t$, choose $y \leq x$ in $V_{n,t}$; then the top t lines of the output of x carry ones. So on every 0-1 input the top t lines end with $\min(w, t)$ ones, and applying this, for every threshold θ, to the input with a 1 exactly where x is at least θ shows that the t largest values of any input end on the top t lines. □

Lemma 2.2. *$\hat{U}_t(n)$ is the least number of generalized comparators in a network that maps $V_{n,t}$ to a single vector.*

Proof. A (t, n)-selector maps $V_{n,t}$ to $\mathbf{1}_{\text{top}}$ by Lemma 2.1. Conversely, untangle a generalized network that maps $V_{n,t}$ to a single vector. The resulting standard network maps $V_{n,t}$ to a single vector as well. Every standard comparator fixes $\mathbf{1}_{\text{top}} \in V_{n,t}$, so that vector is $\mathbf{1}_{\text{top}}$, and Lemma 2.1 applies. □

Thus $\hat{U}_t(n)$ is the length of a shortest path from $V_{n,t}$ to a singleton, where a generalized comparator (a, b) acts on a set S of vectors by moving, in each vector, a 1 on line a to line b when line b holds a 0. The distance from S to a singleton does not change when the lines are permuted. For a set S of vectors of weight t, let $\text{live}(S)$ be the number of lines that carry a 1 in some vector of S.

Lemma 2.3. *A generalized network that maps S to a single vector has at least* $\max(\text{live}(S) - t, \lceil \lg |S| \rceil)$ *comparators.*

Proof. Two vectors with the same image under (a, b) agree outside $\{a, b\}$ and read 10 and 01 on (a, b), so a comparator at most halves $|S|$. Only line a can lose all its ones, so a comparator lowers live by at most one, and the final singleton has live $= t$. □

3. Upper bounds

Yao's recursion [16, Thm. 2.7] compares the inputs in pairs, finds the largest of the $\lfloor n/2 \rfloor$ smaller elements of the pairs, and selects the three largest among the larger elements of the pairs, the unpaired input if n is odd, and that largest smaller element. At most one smaller element is among the three largest inputs, since the larger element of its pair is larger still, and if there is one it is the largest smaller element: a larger one would bring a fourth value with its partner. Hence

$$\hat{U}_3(n) \leq \hat{U}_3(\lceil n/2 \rceil + 1) + 2\lfloor n/2 \rfloor - 1,$$

and starting from $\hat{U}_3(3) = 0$ it yields the upper bound in (1) (Yao starts from $\hat{U}_3(5)$ and $\hat{U}_3(6)$). We untangled the recursion into standard networks for $4 \le n \le 18$, with $3, 6, 8, 11, 13, 15, 17, 20, 22, 24, 26, 28, 30, 32, 34$ comparators. They are in the supplement [13], and each was checked on all 2^n inputs of 0s and 1s by three independent programs.

For $t = 4$ we know Green's construction only through Knuth's answer, which shows it for $n = 11$. The searches of Section 6 output optimal $(4, n)$-selectors for $8 \le n \le 12$, and an independent checker verified one for each n on all 2^n inputs. One for $n = 8$ is

$$[0{:}4]\ [6{:}7]\ [0{:}5]\ [1{:}2]\ [4{:}6]\ [0{:}4]\ [2{:}6]\ [1{:}5]\ [5{:}7]\ [3{:}5]\ [2{:}3]\ [3{:}5]\ [1{:}3]\ [3{:}4],$$

which moves the four largest of eight inputs to the lines $4, \dots, 7$.

4. The step lemma

Let N be a standard network on n lines. Give line i a value smaller than all other inputs; this minimum stays the minimum, and each comparator that meets it moves it to its lower line. Call the comparators that meet it the *path* of i, and its length $L(i)$.

Lemma 4.1. *If N is a (t, n)-selector with $t \ge 2$ and $n \ge t + 1$, then $L(i) \ge 2$ for some line i.*

Proof. Suppose $L(i) \le 1$ for every line i.

(a) Every line is used. If line l is not, its output is its input. The input with a single 1 on line l has weight $1 \le t$, so $l \ge n - t$, and the input with a single 0 on line l has weight $n - 1 \ge t$, so its 0, the minimum, must end below $n - t$, and $l < n - t$.

(b) Let c be the first comparator on line i. It meets the minimum entered on i, which then sits on the lower line of c. As $L(i) = 1$, no later comparator uses that line. We say that c *kills* its lower line. The minimum entered on i ends there, so the killed line lies below $n - t$.

(c) No comparator $c = [a{:}d]$ is the first comparator on both of its lines. Otherwise line a is never used after c by (b), its output is the minimum of the inputs on a and d, and the input with ones on a and d, of weight $2 \le t$, ends with a 1 on line $a < n - t$.

(d) Distinct comparators kill distinct lines: if c comes before c' and both kill line a, then c' uses a after c, against (b).

By (a), (c) and (d), the first comparators of the n lines are n distinct comparators that kill n distinct lines, all below $n - t < n$. This is impossible. □

Proof of Lemma 1.4. Let N be an optimal (t, n)-selector and choose i with $L(i) \ge 2$ by Lemma 4.1. On inputs in which line i carries the minimum, a comparator of the path that receives the minimum on its lower line changes nothing, and one that receives it on its upper line exchanges its two values. Delete the comparators of the first kind, and move the exchanges to the end of the network by renaming the lines of the later comparators. What remains is a generalized network G with $|N| - L(i)$ comparators that never uses line i, followed by a permutation of the lines. Since $n - 1 \ge t$, the minimum is not among the t largest values, so G restricted to the other $n - 1$ lines maps the t largest of its inputs to t fixed lines. By Lemma 2.2, $\hat{U}_t(n - 1) \le |N| - L(i) \le \hat{U}_t(n) - 2$. □

Remark 4.2. For $t \ge 4$, a step $\hat{U}_t(n) \ge \hat{U}_t(n - 1) + 3$ cannot follow from minimum paths alone: there are (t, n)-selectors in which the minimum, entered on any line, meets exactly two comparators. Let $n = 2m$ with $m \ge t$, and use generalized comparators; untangling gives a standard network with the same size and the same minimum paths. (1) Apply $(2i, 2i + 1)$ for $0 \le i < m$, and call $2i$ a *zero line* and $2i + 1$ a *free line*. (2) For each zero line z in turn, sort the m free lines among themselves by a bubble sort, so that the free line h of rank $m - t + 1$ holds a 1 exactly when the free lines hold at least t ones, and then apply (z, h). (3) Sort the free lines once more and take the top t of them as the outputs. The network has $2m + (m + 1)\binom{m}{2}$ comparators.

The minimum entered on line $2i$ or $2i+1$ meets $(2i, 2i+1)$ and moves to the zero line $2i$, which no comparator uses before $(2i, h)$. It meets this second comparator and stays on $2i$: on the input with this single 0 the free lines hold $m \geq t$ ones, so h holds a 1. After that, line $2i$ is never used again.

The network selects. As in Lemma 2.1, it suffices to consider a 0-1 input of weight w. Comparators among the free lines permute their values, and (z, h) changes values only when z holds a 1 and h a 0, in which case it moves that 1 to the free side; so the number of ones on the free lines never decreases. If $w \leq t$, then whenever a zero line z still holds a 1 at its comparator (z, h), the free lines hold at most $w - 1 \leq t - 1$ ones, so h holds a 0 and the 1 moves to h. Every zero line then ends with a 0, all w ones lie on free lines, and step (3) puts them on the outputs. If $w \geq t$, either some comparator (z, h) finds at least t ones on the free lines, which stay there, or every 1 of the zero lines moves to the free side; either way the free lines end with at least t ones, and step (3) puts ones on all t outputs. We also checked these networks on all inputs for $t = 4$ with $n = 10, 12, 14$ (with 70, 117 and 182 comparators), for $t = 5$ with $n = 14$, and for $t = 7$ with $n = 16$.

Any proof of Knuth's equality $\hat{U}_4(n) = 3n - 10$ along these lines must therefore use the size of the network (Proposition 8.4).

5. Weights in time order: bounds for all n

We run a network on all 0-1 inputs at once and keep one number for each line. At every moment a line carries a monotone Boolean function f of the input, and a comparator replaces the functions f and g on its two lines by $f \wedge g$ on the line that receives the smaller value and $f \vee g$ on the other. Let $w(f)$ be the least number of ones in an input on which f is 1, let $W(f) = \min(w(f), t+1)$, and call $\lceil \lg W(f) \rceil$ the *level* of the line. The *potential* of the network at a given moment is the sum of the levels of its lines. Generalized comparators are allowed, and so are output lines other than the top t lines: the potential does not depend on which line of a comparator receives the smaller value, and the proof of Theorem A below uses the output lines only through Lemma 5.1(c). Hence that proof applies verbatim to every generalized network that moves the t largest values to t fixed lines, and by Lemma 2.2 the least size of such a network is $\hat{U}_t(n)$.

Lemma 5.1. (a) *If $W(f) = a \leq b = W(g)$, then $W(f \vee g) = a$ and $b \leq W(f \wedge g) \leq \min(a + b, t + 1)$.*

(b) *Every comparator raises the potential by at most* 1.

(c) *At the end of a (t, n)-selector with $n \geq t$, every line other than the t output lines has $W = t + 1$, and every output line has $W \leq t$.*

Proof. (a) The function $f \vee g$ is 1 exactly where f or g is, so $w(f \vee g) = \min(w(f), w(g))$. An input on which $f \wedge g$ is 1 makes both functions 1, so $w(f \wedge g) \geq \max(w(f), w(g))$. If x and y are inputs of weights $w(f)$ and $w(g)$ with $f(x) = g(y) = 1$, then both functions are 1 on $x \vee y$, since they are monotone, so $w(f \wedge g) \leq w(f) + w(g)$. The cap at $t + 1$ keeps these relations. (b) By (a) the new levels are $\lceil \lg a \rceil$ and $\lceil \lg c \rceil$ for some c with $b \leq c \leq \min(2b, t + 1)$, so the potential grows by $\lceil \lg c \rceil - \lceil \lg b \rceil \leq \lceil \lg 2b \rceil - \lceil \lg b \rceil = 1$. (c) On an input with at most t ones, all the ones are among the t largest values and end on the output lines, so every other line holds 0; hence $w \geq t + 1$ on those lines. On an input with exactly t ones every output line holds 1, so $w \leq t$ on the output lines. □

At the start every line carries one input variable, of weight 1 and level 0. Call the amount by which a comparator falls short of raising the potential by 1 its *waste*. For a (t, n)-selector with s

comparators, let σ be the total waste and ε the sum of the levels of the output lines at the end. Lemma 5.1 then gives

$$s = (n-t)\lceil \lg(t+1)\rceil + \sigma + \varepsilon, \qquad \sigma \ge 0,\ \varepsilon \ge 0, \tag{2}$$

and with it Alekseev's bound.

Lemma 5.1(a) describes a comparator by the multiset of the weights of its two lines: they change from $a \le b$ to a and c, where $b \le c \le \min(a+b, t+1) \le 2b$. We say that the comparator *lifts* the weight b to c, and call a its *helper*; if $a = b$, either weight may be taken as the lifted one. The comparator raises the potential by $\lceil \lg c\rceil - \lceil \lg b\rceil$, which is 0 or 1; we call this its *gain*, so that its waste is 1 minus its gain. These are statements about weights, not lines: the line that carried b may receive $f \vee g$ and leave the comparator with the smaller weight a. We therefore follow each of the n weights through the network as an entry of the multiset, which starts at 1 and changes only when it is lifted.

For $j \ge 1$ let $Q_j = \{2^{j-1}+1, \dots, 3\cdot 2^{j-1}\}$, so that $Q_1 = \{2,3\}$, $Q_2 = \{3,\dots,6\}$ and $Q_3 = \{5,\dots,12\}$. The weights of level j, from $2^{j-1}+1$ to 2^j, lie in Q_j, and Q_i and Q_j are disjoint if $|i-j| \ge 2$. A comparator *empties* Q_j if some line has a weight in Q_j before it and no line has one after it.

Lemma 5.2. (a) *If a comparator of gain* 1 *lifts a weight* $b \in Q_j$ *to a weight* $c \notin Q_j$*, then its helper lies in* Q_j*.*

(b) *A comparator empties at most one of the sets* Q_j*.*

(c) *A weight* $v \ge 2$ *lies in at most* $\lceil \lg v\rceil$ *of the sets* Q_j*.*

Proof. (a) Let a be the helper. Since $c \ge b > 2^{j-1}$ and $c \notin Q_j$, we have $c > 3\cdot 2^{j-1}$. If $b \le 2^j$, then $a \ge c - b > 3\cdot 2^{j-1} - 2^j = 2^{j-1}$. Otherwise b has level $j+1$, so c has level $j+2$, as the gain is 1; then $c > 2^{j+1}$ and $a \ge c - b > 2^{j+1} - 3\cdot 2^{j-1} = 2^{j-1}$. In both cases $2^{j-1} < a \le b \le 3\cdot 2^{j-1}$. (b) The comparator changes one weight of the multiset, b to c, so it can empty only sets that contain b. At most two sets contain b, and if two do, they are Q_j and Q_{j+1} for some j, and $2^j < b \le 3\cdot 2^{j-1}$. Then $2^j < c \le 2b \le 3\cdot 2^j$, so c lies in Q_{j+1}, and Q_{j+1} is not emptied. (c) The weight v lies in at most two of the sets, and if it lies in Q_j, then $v > 2^{j-1}$, so $\lceil \lg v\rceil \ge j$. So if v lies in two sets, Q_j and Q_{j+1}, then $\lceil \lg v\rceil \ge j+1 \ge 2$, and if it lies in one, then $\lceil \lg v\rceil \ge 1$, as $v \ge 2$. □

Proof of Theorem A. Let $t = 2^k$ with $k \ge 2$, so that the cap $t+1$ has level $k+1$. We use the sets $Q_1, \dots, Q_{k-1}$, which lie in $\{2,\dots,3t/4\}$; by Lemma 5.1(c), a line that has a weight in one of them at the end is an output line. At each moment and for $1 \le j \le k-1$, let $\chi_j = 1$ if some line had a weight in Q_j at an earlier moment and no line has one now, and $\chi_j = 0$ otherwise. Let Φ be the potential plus $\chi_1 + \cdots + \chi_{k-1}$. At the start every weight is 1, so $\Phi = 0$.

Each comparator raises Φ by at most 1. Indeed, χ_j can rise only at a comparator that empties Q_j. If no χ_j rises, Φ rises by at most the gain, which is at most 1. If χ_j rises, the comparator lifts the last weight in Q_j to a weight outside Q_j, and its helper is not in Q_j; so its gain is 0 by Lemma 5.2(a), and no other χ_i rises, by Lemma 5.2(b).

Since $n \ge t+1$, some line has weight $t+1$ at the end, by Lemma 5.1(c). As an entry of the multiset, this weight started at 1, of level 0, and each lift raised its level by at most 1, since $c \le 2b$. So for each $j \le k-1$ it had level j at some moment, and then it lay in Q_j. Hence, at the end, for each $j \le k-1$ either $\chi_j = 1$ or some output line has a weight in Q_j. By Lemma 5.2(c), the weight of an output line lies in at most as many of the sets as its level, so ε, the sum of the levels of the output lines, is at least the number of sets Q_j with $j \le k-1$ that still contain a weight. Therefore $\varepsilon + \chi_1 + \cdots + \chi_{k-1} \ge k-1$, and at the end

$$\Phi = (n-t)(k+1) + \varepsilon + \chi_1 + \cdots + \chi_{k-1} \ge (n-t)(k+1) + k - 1.$$

As Φ starts at 0 and each comparator raises it by at most 1, the network has at least this many comparators. □

In the notation of (2), the proof shows that $\sigma \geq \chi_1 + \cdots + \chi_{k-1}$ at the end, and hence $\sigma + \varepsilon \geq k - 1$.

The proof uses only the weights and Lemma 5.1, so it holds in the following *weight model*. A state is a multiset of n weights, at the start all equal to 1; a move replaces two weights $a \leq b$ by a and some c with $b \leq c \leq \min(a + b, t + 1)$; and a final state has $n - t$ weights equal to $t + 1$ and t weights at most t. Gain, waste, σ, ε and Φ are defined as before. By Lemma 5.1, every (t, n)-selector gives a sequence of as many moves from the start to a final state, with the same σ and ε, and the proof of Theorem A shows that $\sigma + \varepsilon \geq k - 1$ for every such sequence.

To go further, fix such a sequence, for $t = 2^k$ with $k \geq 2$, and let F be the multiset of its t output weights at the end. By the proof of Theorem A, every set Q_j with $j \leq k - 1$ has been entered by the end, and at the end these sets hold only output weights. Let D be the total amount by which the moves fall short of raising Φ by 1, and let $q(F)$ be the number of sets Q_j, $j \leq k - 1$, that contain a weight of F. Then Φ ends at $(n - t)(k + 1) + \varepsilon + k - 1 - q(F)$, the number of moves is this value plus D, and (2) gives

$$\sigma + \varepsilon = k - 1 + f(F) + D, \qquad \text{where} \quad f(F) = \sum_{v \in F} \lceil \lg v \rceil - q(F) \geq 0 \tag{3}$$

by Lemma 5.2(c). Call a move *idle* if it has gain 0 and empties no set Q_j and enters none; such a move adds 1 to D. For $1 \leq j \leq k - 1$ let $S_j = \{3 \cdot 2^{j-1} + 1, \ldots, 2^{j+1}\}$, the upper part of level $j + 1$. These sets are disjoint; S_j lies in Q_{j+1} if $j \leq k - 2$, and S_{k-1} meets none of the sets $Q_1, \ldots, Q_{k-1}$. So a move that lifts a weight of S_j to a weight of S_j is idle.

Lemma 5.3. *Let $1 \leq j \leq k - 1$, suppose that no output weight lies in $Q_j \cup S_j$, and let A be the largest output weight with $A \leq 2^{j-1}$. Then A exists, and at least* $\max\bigl(0, \lceil (2^{j-1} + 1)/A \rceil - 2\bigr)$ *moves lift a weight of S_j to a weight of S_j.*

Proof. The set Q_j has been entered and holds no output weight, so it is empty at the end. Let τ be the last move that empties it; after τ no weight enters Q_j. The move τ lifts the last weight b of Q_j to some $c \notin Q_j$, and its helper a is not in Q_j. Then $c > 3 \cdot 2^{j-1}$, and $a \leq 2^{j-1}$, as $a \leq b \leq 3 \cdot 2^{j-1}$ and $a \notin Q_j$; so $c \leq a + b \leq 2^{j+1}$, and $c \in S_j$.

After τ, a weight $w \leq 2^{j-1}$ stays at most 2^{j-1}, since a lift takes it to at most $2w \leq 3 \cdot 2^{j-1}$ and the weights from $2^{j-1} + 1$ to $3 \cdot 2^{j-1}$ form Q_j. So it ends as an output weight, and it is at most A at every moment after τ. The helper a is such a weight, so A exists and $a \leq A$. It follows that after τ no weight enters S_j, as it would have to come from Q_j or from a weight at most 2^{j-1}; and as S_j holds no output weight and $2^{j+1} \leq t$, it is empty at the end.

While S_j is not empty after τ, let m be its least weight; m never decreases. A weight m can be lifted only with a helper at most m, which is a weight at most A or another weight m. So m changes only at a move that lifts the last weight m, with a helper at most A, to at most $m + A$. If $m \leq 2^{j+1} - A$, the lifted weight stays in S_j, so the move lifts a weight of S_j to a weight of S_j, and m rises by at most A. Just after τ, $m \leq c \leq 3 \cdot 2^{j-1} + A$, and m exceeds $2^{j+1} - A$ before S_j becomes empty, since the last weight to leave S_j is lifted with a helper at most A. Hence m rises at least $\lceil (2^{j-1} + 1 - 2A)/A \rceil = \lceil (2^{j-1} + 1)/A \rceil - 2$ times while $m \leq 2^{j+1} - A$, each time at a move that lifts a weight of S_j to a weight of S_j. □

For a multiset F of weights in $\{1, \ldots, t\}$, let $J(F)$ be the set of indices $j \leq k - 1$ such that no weight of F lies in $Q_j \cup S_j$, and for $j \in J(F)$ let $A_j(F)$ be the largest weight of F that is at

most 2^{j-1}. Let

$$\mathrm{LB}(k) = \min_F \Big(f(F) + \sum_{j \in J(F)} \max\big(0, \lceil (2^{j-1}+1)/A_j(F) \rceil - 2\big) \Big), \tag{4}$$

the minimum over the multisets F of t weights in $\{1, \dots, t\}$ for which $A_j(F)$ exists for every $j \in J(F)$.

Proof of Theorem C. Let F be the output weights of a sequence of moves of the weight model to a final state. By Lemma 5.3, $A_j(F)$ exists for every $j \in J(F)$, and at least $\max(0, \lceil (2^{j-1}+1)/A_j(F) \rceil - 2)$ idle moves lift a weight of S_j to a weight of S_j. These moves are distinct for distinct j, as the sets S_j are disjoint, and each adds 1 to D. By (3), $\sigma + \varepsilon \ge k - 1 + \mathrm{LB}(k)$, and Lemma 5.1 turns this into the bound on $\hat{U}_t(n)$. □

Proof of Theorem B. Let $k \ge 4$. By the proof of Theorem C it suffices to show that the expression minimized in (4) is at least $k-4$ for every F; we use only its term $j = k-1$. For a weight v let $d(v)$ be $\lceil \lg v \rceil$ minus the number of sets Q_j, $j \le k-1$, that contain v; then $f(F) \ge \sum_{v \in F} d(v)$, and each term is nonnegative by Lemma 5.2(c). We have $d(1) = d(2) = 0$. If $2^{i-1} < v \le 2^i$ with $2 \le i \le k-1$, then $d(v) = i-2$ if $v \le 3 \cdot 2^{i-2}$, and $d(v) = i-1$ otherwise; and if $2^{k-1} < v \le 2^k$, then $d(v) = k-1$ if $v \le 3 \cdot 2^{k-2}$, and $d(v) = k$ otherwise.

Let $h = 2^{k-2}$, so that $Q_{k-1} = \{h+1, \dots, 3h\}$ and $S_{k-1} = \{3h+1, \dots, 4h\}$. If F has a weight in Q_{k-1}, this weight has $d \ge k-3$, and if F has a weight in S_{k-1}, it has $d = k$. Otherwise $k-1 \in J(F)$, and with $A = A_{k-1}(F) \le h$ the expression is at least $d(A) + \max(0, \lceil (h+1)/A \rceil - 2)$. If $A = 1$, this is $h - 1 \ge k-4$. Otherwise $2^{i-1} < A \le 2^i$ for some i with $1 \le i \le k-2$; let $e = k-i \ge 2$. If $A \le 3 \cdot 2^{i-2}$, then $d(A) = k-e-2$ and $(h+1)/A > 2^e/3$, and $k-e-2+\lceil 2^e/3 \rceil - 2$ is $k-4$ for $e = 2, 3$ and at least $k-2$ for $e \ge 4$. If $A > 3 \cdot 2^{i-2}$, then $d(A) = k-e-1$ and $(h+1)/A > 2^{e-2}$, so that the expression is at least $k - e - 1 + 2^{e-2} - 1 \ge k-3$. □

Remark 5.4. In (4), $f(F)$, $J(F)$ and the comparisons with 2^{j-1} depend on a weight only through the half of its level in which it lies, and each term of the sum decreases as $A_j(F)$ grows; so only the largest weight of F in each half-level matters, and $\mathrm{LB}(k)$ is a minimum over finitely many sets. Two programs in the supplement [13], written separately, compute it; these values are computed, not formalized:

k	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\mathrm{LB}(k)$	0	0	0	1	2	3	4	6	8	10	12	15	18	21	24	28	32	36	40

So $\mathrm{LB}(k) = \max(0, k-4)$ for $k \le 8$, and for $9 \le k \le 20$ Theorem C is stronger than Theorem B, by $\mathrm{LB}(k) - (k-4)$, which grows from 1 to 24 in the table; for $t = 512$ it gives $\hat{U}_t(n) \ge 10n - 5106$. For $k \ge 2$, Corollary D holds as well with $\mathrm{LB}(k)$ in place of δ_k, by the same proof.

We write $m \times (a, b \to c)$ for m moves that each lift a weight b to c with a helper a.

Proposition 5.5. *Let $t = 2^k$. In the weight model, the least value of $\sigma + \varepsilon$ over all sequences of moves to a final state is $k-1$ for $k = 2, 3, 4$, for every $n \ge 5$, 9 and 18 respectively, 5 for $k = 5$ and every $n \ge 34$, and 7 for $k = 6$ and every $n \ge 67$.*

Proof. Theorems A and B give the lower bounds. The following sequences reach them; in each, one checks move by move that the helpers are present and every move is allowed, and that the last state is a final state. For $k = 2$: $(n-3) \times (1, 1 \to 2)$, $(n-4) \times (2, 2 \to 4)$ and $(n-4) \times (1, 4 \to 5)$, with no waste and the output weights $1, 1, 1, 2$. For $k = 3$: $(n-7) \times (1, 1 \to 2)$, $(n-7) \times (1, 2 \to 3)$, $(n-8) \times (3, 3 \to 6)$ and $(n-8) \times (3, 6 \to 9)$, with no waste and the output weights seven times 1 and once 3. For $k = 4$: $(n-15) \times (1, 1 \to 2)$, $2 \times (2, 2 \to 4)$, $(n-17) \times (1, 2 \to 3)$, $(4, 4 \to 8)$, $(3, 4 \to 6)$, $(n-18) \times (3, 3 \to 6)$, $(n-17) \times (3, 6 \to 9)$, $(n-17) \times (8, 9 \to 17)$, and $(3, 8 \to 11)$, $(3, 11 \to 14)$, $(3, 14 \to 17)$, with waste 1 (the move $11 \to 14$) and the output weights fifteen times

1 and once 3. For $k = 5$: $(n-30) \times (1, 1 \to 2)$, $3 \times (2, 2 \to 4)$, $2 \times (4, 4 \to 8)$ and $(8, 8 \to 16)$; then, $n - 34$ times, the moves $(1, 2 \to 3)$, $(2, 3 \to 5)$, $(4, 5 \to 9)$, $(8, 9 \to 17)$, $(16, 17 \to 33)$ on one weight 2; and $(2, 4 \to 6)$, $(6, 8 \to 11)$, $(6, 11 \to 17)$, $(16, 17 \to 33)$, $(6, 16 \to 22)$, $(6, 22 \to 28)$, $(6, 28 \to 33)$, with waste 1 (the move $22 \to 28$) and the output weights thirty times 1, once 2 and once 6, so that $\varepsilon = 4$. For $k = 6$: $(n-62) \times (1, 1 \to 2)$, $4 \times (2, 2 \to 4)$, $(1, 2 \to 3)$, $3 \times (4, 4 \to 8)$, $(3, 4 \to 6)$, $(3, 6 \to 9)$, $2 \times (8, 8 \to 16)$, $(8, 9 \to 17)$ and $(16, 16 \to 32)$; then, $n - 67$ times, the moves $(1, 2 \to 3)$, $(3, 3 \to 6)$, $(3, 6 \to 9)$, $(8, 9 \to 17)$, $(16, 17 \to 33)$, $(32, 33 \to 65)$ on one weight 2; and $(3, 8 \to 11)$, $(16, 17 \to 33)$, $(11, 16 \to 22)$, $(11, 22 \to 33)$, $2 \times (32, 33 \to 65)$, $(11, 32 \to 43)$, $(11, 43 \to 54)$, $(11, 54 \to 65)$, with waste 1 (the move $43 \to 54$) and the output weights sixty-two times 1, once 3 and once 11. □

Hence, at the sizes n of Proposition 5.5, no argument that uses only these properties of the weights can improve Theorem A for $t = 4$, 8 and 16, or Theorem B for $t = 32$ and $t = 64$. For $k \geq 7$ we know only the lower bound $k - 1 + \mathrm{LB}(k)$ of Theorem C.

The pruning step behind Corollary D is Yao's. His selectors collect the t smallest values, and his proof of $\hat{U}_3(n) \geq 2n - 6 + \lceil \lg \lceil n/3 \rceil \rceil$ follows the smallest value and prunes a $(3, n)$-selector to a $(2, n-1)$-selector [16, proof of Thm. 2.6]; he notes that the argument extends by induction to every t that is not a power of two [16, Sec. 2.3.3]. In our orientation it follows the largest value.

Lemma 5.6. *If $t \geq 2$ and $n \geq t + 1$, then $\hat{U}_t(n) \geq \hat{U}_{t-1}(n-1) + \lceil \lg \lceil n/t \rceil \rceil$.*

Proof. Let N be an optimal (t, n)-selector. Give line i a value larger than all other inputs; this maximum stays the maximum, each comparator that meets it moves it to its upper line, and it ends on an output line. Call the comparators that meet it the *upper path* of i, and its length $L^+(i)$. The upper paths of the n lines end on the t output lines, so at least $\lceil n/t \rceil$ of them end on the same output line. Two upper paths that meet at a comparator coincide from there on, so these paths form a binary tree, with the comparators on them as inner nodes and the input lines as leaves, and $L^+(i) \geq \lceil \lg \lceil n/t \rceil \rceil$ for one of these lines i. Now argue as in the proof of Lemma 1.4, with the maximum in place of the minimum: deleting the upper path of i and renaming the lines of the later comparators leaves a generalized network with $|N| - L^+(i)$ comparators that never uses line i, followed by a permutation of the lines that takes line i to an output line, and on the other $n - 1$ lines this network moves the $t - 1$ largest of its inputs to $t - 1$ fixed lines. By Lemma 2.2, $\hat{U}_{t-1}(n-1) \leq |N| - L^+(i) \leq \hat{U}_t(n) - \lceil \lg \lceil n/t \rceil \rceil$. □

Proof of Corollary D. Let $m = n - t + 2^k$, so that $m \geq 2^k + 1$. Lemma 5.6 is an inequality between least sizes: as in the proof of Lemma 1.4, the pruned network, which has generalized comparators and new output lines, is compared with $\hat{U}_{t-1}(n-1)$ through Lemma 2.2. So the lemma can be applied repeatedly, and at $(t - i, n - i)$ for $i = 0, \ldots, t - 2^k - 1$ it gives

$$\hat{U}_t(n) \geq \hat{U}_{2^k}(m) + \sum_{i=0}^{t-2^k-1} \lceil \lg \lceil \tfrac{n-i}{t-i} \rceil \rceil.$$

If $2 \leq k \leq 4$, Theorem A gives $\hat{U}_{2^k}(m) \geq (k+1)(m - 2^k) + k - 1$, if $k \geq 5$, Theorem B gives $\hat{U}_{2^k}(m) \geq (k+1)(m - 2^k) + 2k - 5$, and if $k = 1$, Alekseev's bound gives $\hat{U}_2(m) \geq 2(m-2)$; in each case this is $(k+1)(n-t) + k - 1 + \delta_k$. This proves the first inequality. For the second, each term of the sum is at least 1, since $n - i > t - i$, and $k + 1 = \lceil \lg(t+1) \rceil$, since $2^k \leq t < 2^{k+1}$. By Lemma 2.2, both bounds hold as well for generalized networks that move the t largest values to any t fixed lines. □

Corollary 5.7. *$\hat{U}_4(n) \geq 3n - 11$ for every $n \geq 5$, with equality at $n = 5$. Hence $\hat{U}_4(n)$ is $3n - 11$ or $3n - 10$ for every $n \geq 6$.*

Proof. The bound is Theorem A for $t = 4$, and $\hat{U}_4(5) = \hat{U}_1(5) = 4$. For $n \geq 6$ the bound of Knuth's exercise [7, answer to exercise 5.3.4–20(b)] gives $\hat{U}_4(n) \leq 3n - 10$. □

By (2), Knuth's expected equality $\hat{U}_4(n) = 3n - 10$ says that $\sigma + \varepsilon \geq 2$ for every $(4, n)$-selector with $n \geq 6$, while the proof of Theorem A gives $\sigma + \varepsilon \geq 1$, and by Proposition 5.5 the weight model allows $\sigma + \varepsilon = 1$. Theorem 1.3 proves $\sigma + \varepsilon \geq 2$ for $6 \leq n \leq 12$; the step from 1 to 2 uses structure of the networks that the weights do not record.

Remark 5.8. The proof of Theorem A works for every $t \geq 2$ with the sets Q_j for $1 \leq j \leq \lfloor \lg(2t/3) \rfloor$, which are the sets Q_j inside $\{2, \ldots, t\}$. Lemma 5.2 holds for every t, and a weight that ends at $t + 1$ passes through every level up to $\lceil \lg(t + 1) \rceil$, which exceeds $\lfloor \lg(2t/3) \rfloor$. A single potential therefore gives $\hat{U}_t(n) \geq (n - t)\lceil \lg(t + 1) \rceil + \lfloor \lg(2t/3) \rfloor$ for $t \geq 2$ and $n \geq t + 1$. It equals the second bound of Corollary D for $t = 2, 3, 4, 8, 16$, and is weaker for every other t.

6. Exhaustive search

By Lemmas 2.2 and 2.3, $\hat{U}_t(n) > D$ follows from a breadth-first search over the sets reachable from $V_{n,t}$: layer k holds the states reachable with k generalized comparators, up to a permutation of the lines, and a state in layer k is discarded when $k + \max(\mathrm{live}(S) - t, \lceil \lg |S| \rceil) > D$. If no layer up to D contains a singleton, then $\hat{U}_t(n) > D$. Two states may be merged only if one is a permutation of the other, and each program uses a merge test that never merges any other states.

Two programs, written independently, compute $\hat{U}_t(n)$ this way. We call them B and C, after their folders in the supplement [13], which also holds their logs. Program B merges states that are permutations of each other after sorting the lines by an invariant. Program C was written from the definition alone, without access to Program B; it uses its own canonical forms, the bound of Lemma 2.3, and a checkpointed, multithreaded variant for the largest cases. Table 2 lists the values each program gives. For $\hat{U}_4(12)$, Program B's search ran under a new *parallel driver*, which runs it on several threads with the same states, merge test and moves as Program B. We compared the driver's output at every depth with that of the sequential program for $t = 3$ with $10 \leq n \leq 12$ and for $t = 4$ with $8 \leq n \leq 11$; the two agree.

value	programs	largest run
$\hat{U}_2(n) = 2n - 4$, $3 \leq n \leq 10$	C, and B for $n = 8, 9$	—
$\hat{U}_3(n)$, $4 \leq n \leq 10$	C, and B for $6 \leq n \leq 10$	—
$\hat{U}_3(11) = 20$	B, C	B: 7.2 s, 314,518 states; C: 25.4 s
$\hat{U}_3(12) = 22$	B, C	B: 74.9 s, 2,349,706 states; C: 243.4 s, largest layer 460,255
$\hat{U}_3(13) = 24$	B	2,131.5 s, 18,962,336 states
$\hat{U}_4(n)$, $n = 8, 9, 10$	B, C	at $n = 10$, B: 34.1 s, 1,553,147 states; C: 199.2 s, largest layer 397,180
$\hat{U}_4(11) = 23$	B, C	B: 3,421.5 s, 48,814,192 states; C: 149.1 s on 90 threads, 45,358,610 states
$\hat{U}_4(12) = 26$	B, C	B: 868.9 s on 190 threads, 1,908,917,869 states; C: 7,723.4 s on 90 threads, 1,806,537,098 states, largest layer 486,241,921

Table 2. The searches, with the figures of the runs logged in the supplement. Times are for one core unless stated. The two programs count states differently.

For the values used in the proofs of Theorems 1.1 and 1.2, and for $\hat{U}_4(n)$ with $6 \le n \le 10$, a separate generator writes the states of the search, with their depths, as a certificate. A certificate is *closed* if it contains $V_{n,t}$ at depth 0, no state is a singleton, and every move from a state of depth $d < D$ leaves the state unchanged, or leaves a set that needs more than $D - d - 1$ further comparators by Lemma 2.3, or gives a relabeling of a state of depth at most $d + 1$. A closed certificate proves $\hat{U}_t(n) > D$. The certificate for $\hat{U}_3(7) > 10$ has 191 states, and the one for $\hat{U}_3(11) > 19$ has 210,535. A separate program, which shares no code with the generator, checks that the certificates are closed.

7. Proofs of the theorems

Proof of Theorem 1.1. Write $Y(n) = 2n - 5 + \lfloor \lg(n-3) \rfloor$. Then $Y(n) - Y(n-1) = 3$ if $n-3$ is a power of two, and 2 otherwise. We have $\hat{U}_3(4) = \hat{U}_1(4) = 3 = Y(4)$ and $\hat{U}_3(5) = \hat{U}_2(5) = 6 = Y(5)$. The searches show that $\hat{U}_3(7) \ge 11 = Y(7)$ and $\hat{U}_3(11) \ge 20 = Y(11)$. By Lemma 1.4, $\hat{U}_3(n) \ge \hat{U}_3(n-1) + 2$, which covers every other $n \le 18$: the next size with $n - 3$ a power of two is 19. So $\hat{U}_3(n) \ge Y(n)$ for $4 \le n \le 18$, and Yao's networks show that equality holds. The searches of Table 2 agree for $n \le 13$. □

Proof of Theorem 1.2. $\hat{U}_3(n) \ge \hat{U}_3(11) + 2(n - 11) = 2n - 2$ by Lemma 1.4. □

Proof of Theorem 1.3. For $n = 6, 7$ see Section 1. For $8 \le n \le 12$ Programs B and C find no $(4, n)$-selector with fewer than $3n - 10$ comparators (for $n = 12$, Program B under its parallel driver), and the networks of Section 3 have $3n - 10$ comparators. □

Thus $\hat{U}_4(n) = 8, 11, 14, 17, 20, 23, 26$ for $n = 6, \ldots, 12$: Knuth's bound $3n - 10$, one more than the bound of Theorem A and two more than Alekseev's bound $3(n - 4)$.

8. The sizes $2^j + 3$

Proposition 8.1. *Let $M \ge 4$. Then $\hat{U}_3(n) = 2n - 5 + \lfloor \lg(n-3) \rfloor$ for all $4 \le n \le M$ if and only if $\hat{U}_3(m) \ge \hat{U}_3(m-1) + 3$ for every $m = 2^j + 3$ with $5 \le m \le M$.*

Proof. The base is $\hat{U}_3(4) = \hat{U}_1(4) = 3 = Y(4)$. For $n \ge 5$, $Y(n) - Y(n-1)$ is 3 at the sizes $m = 2^j + 3$ and 2 otherwise, and by Lemma 1.4, $\hat{U}_3(n) - \hat{U}_3(n-1) \ge 2$. If the step is at least 3 at every such $m \le M$, induction from $n = 4$ yields $\hat{U}_3(n) \ge Y(n)$ for $4 \le n \le M$; conversely, if $\hat{U}_3 = Y$ on $4 \le n \le M$, the step at each such m is 3. The upper bound $\hat{U}_3(n) \le Y(n)$ for all $n \ge 4$ is Yao's construction [16, Thm. 2.7]; we checked the networks explicitly only for $n \le 18$ (Section 3). □

Yao's question is therefore whether $\hat{U}_3$ grows by three at every size $2^j + 3$. This holds for $m = 5, 7, 11$. At $m = 19$ we know only $\hat{U}_3(19) \in \{36, 37\}$, and the number of states that Program B keeps grows by a factor of about 8 from $n = 12$ to $n = 13$ (Table 2), which puts the search out of reach at $n = 19$.

Suppose $\hat{U}_3(m) = \hat{U}_3(m-1) + 2$ for some $m = 2^j + 3$, and let N be an optimal $(3, m)$-selector. By the proof of Lemma 1.4, every minimum path of N has length at most 2, some has length 2, and deleting it leaves an optimal $(3, m-1)$-selector N'. By Lemma 2.2 we may treat N and N' as generalized networks. After renaming lines, N then arises from N' by *inserting* a line z with two comparators. We number the comparators of N' by *times* $1, 2, \ldots$; then z meets a line x just before the comparator at time p_1 of N' and a line y just before time $p_2 \ge p_1$, and at both new comparators z keeps the smaller value. Let O' be the set of the three lines on which N' puts the three largest values. The following criterion decides whether an insertion gives a selector.

A *slot* of N' is a pair $s = (l, \tau)$ of a line l and a time τ, meaning line l just before the comparator at time τ (or at the end). For a pair $\{p, q\}$ of lines, run N' on the input with ones on p and q. Let $\mathrm{Ones}(s)$ be the set of pairs for which line l holds a 1 at time τ. A pair outside $\mathrm{Ones}(s)$ is *good* at s if an extra 1 put on line l at time τ ends, together with the other two, on the lines O', and *bad* otherwise. Let $\mathrm{Good}(s)$ be the set of pairs good at s.

Proposition 8.2 (slot criterion)**.** *The insertion of z at the slots (x, p_1) and (y, p_2) gives a $(3, m)$-selector if and only if no pair is bad at (x, p_1) and $\mathrm{Ones}(x, p_1) \subseteq \mathrm{Good}(y, p_2)$.*

Proof. By Lemmas 2.1 and 2.2 only inputs of weight 3 matter. If z holds a 0, both new comparators leave the other value in place and the network runs as N'. Otherwise the input is z together with a pair $\{p, q\}$, and N' runs on $\{p, q\}$ undisturbed until z meets a line holding a 0. If line x holds a 0 at time p_1, the 1 of z moves to x there and the run succeeds if and only if the pair is good at (x, p_1). If line x holds a 1, z keeps its 1 until time p_2. If line y then holds a 0, the run succeeds if and only if the pair is good at (y, p_2). If line y holds a 1, the 1 of z stays on z to the end, and z is not one of the three output lines: the minimum of any input, entered on z, ends on z, and it is not among the three largest. These three cases are exhaustive. □

Hence $\hat{U}_3(2^j + 3) \geq \hat{U}_3(2^j + 2) + 3$ if and only if, in every optimal $(3, 2^j + 2)$-selector, for every slot (x, p_1) without bad pairs and every slot (y, p_2) with $p_2 \geq p_1$, some pair of $\mathrm{Ones}(x, p_1)$ is not good at (y, p_2). We checked this exhaustively at $m = 7$ and on samples at $m = 11$ (Table 3). The margin is one pair: in each of the 7,245 networks at $m = 7$ and of 400 sampled networks at $m = 11$, the least number of pairs of $\mathrm{Ones}(x, p_1)$ that are not good at (y, p_2), over all such pairs of slots, is exactly 1. Between the sizes $2^j + 3$, by contrast, many insertions succeed.

insertions into	networks	insertions	selectors
all optimal $(3, 5)$-selectors	3,450	2,415,000	22,935
all optimal $(3, 6)$-selectors	7,245	11,736,900	0
random optimal $(3, 7)$-selectors	2,000	7,644,000	71,552
random optimal $(3, 10)$-selectors	2,000	34,200,000	0

TABLE 3. Insertions of a line with a minimum path of length 2 into optimal $(3, m-1)$-selectors, counted by brute force; the criterion of Proposition 8.2 gives the same counts for the first two rows. Standard networks are counted up to the order of commuting comparators.

Conjecture 8.3. *For every $j \geq 1$, in every optimal $(3, 2^j + 2)$-selector, for every slot (x, p_1) without bad pairs and every slot (y, p_2) with $p_2 \geq p_1$, some pair of $\mathrm{Ones}(x, p_1)$ is not good at (y, p_2). Equivalently, Yao's upper bound in (1) is exact for all $n \geq 4$.*

The same reduction works for $t = 4$, where it has to use the size of the networks (Remark 4.2).

Proposition 8.4. *$\hat{U}_4(n) = 3n - 10$ for all $n \geq 6$ if and only if, for every $n \geq 8$, every $(4, n)$-selector in which every minimum path has length at most 2 has at least $3n - 10$ comparators. Likewise, Yao's upper bound is exact for all $n \geq 4$ if and only if, for every $m = 2^j + 3 \geq 19$, every $(3, m)$-selector in which every minimum path has length at most 2 has at least $2m - 5 + j$ comparators.*

Proof. One direction is clear. For the other, recall $\hat{U}_4(6) = 8$ and $\hat{U}_4(7) = 11$, and let $n \geq 8$ be the least size with $\hat{U}_4(n) < 3n - 10$. Here the upper bounds are needed: by the bound of Knuth's exercise [7, answer to exercise 5.3.4–20(b)], $\hat{U}_4(n-1) \leq 3(n-1) - 10$, so by minimality

$\hat{U}_4(n-1) = 3n-13$, equal to the formula. Lemma 1.4 then implies $\hat{U}_4(n) \geq 3n-11$, hence $\hat{U}_4(n) = 3n-11 = \hat{U}_4(n-1)+2$. A minimum path of length 3 or more in an optimal $(4,n)$-selector would leave, by the proof of Lemma 1.4, a $(4,n-1)$-selector with fewer than $\hat{U}_4(n-1)$ comparators. So every minimum path of this network has length at most 2, and it has $3n-11$ comparators. For $t=3$, let m be the least size with $\hat{U}_3(m) < Y(m)$. By Proposition 8.1 it has the form 2^j+3, and $m \geq 19$ by Theorem 1.1. By Yao's construction and minimality, $\hat{U}_3(m-1) = Y(m-1) = Y(m)-3$, so $\hat{U}_3(m) = Y(m)-1 = \hat{U}_3(m-1)+2$, and the same argument applies. □

In this restricted class an exact search is cheaper, and two independently written programs give the same results: there is no $(4,5)$- or $(4,6)$-selector with every minimum path of length at most 2, the least one for $(4,7)$ has $11 = 3\cdot 7-10$ comparators, and for $(3,6)$ and $(3,7)$ the least sizes are 8 and 11, as without the restriction.

9. Lead-pattern potentials for selecting three

The lead-pattern potentials of [11] also give lower bounds for selection. We use them only for $t=3$, with P the set of the 0-1 inputs of weights 2 and 3; every statement of this section is for this P. Run a network on all inputs of P at once: each line then carries the set of inputs on which it holds a 1, an up-set of P (in [11], the *pattern* of the wire segment, or *lead*, between two comparators, on which the potentials are defined), and a comparator $[a{:}b]$ replaces the sets f on line a and g on line b by $f\cap g$ and $f\cup g$. A cover of P is a pair $u\subset v$ of an input u of weight 2 and an input v of weight 3; a set f carries it if $v\in f$ and $u\notin f$. At every moment each cover is carried by exactly one line, since the number of lines whose set contains a given input is its weight; so the two lines of a comparator carry disjoint sets of covers, as for merging [11, Lemma 3.3].

Let w be a function on the up-sets of P with $w(\emptyset)=0$ that is invariant under permutations of the lines, and suppose

$$\mathrm{def}_w(f,g) = w(f\cup g)+w(f\cap g)-w(f)-w(g) \leq 1$$

for all incomparable up-sets f,g that carry disjoint sets of covers. Then each comparator raises $\sum_{\text{lines}} w$ by at most 1. At the start line c carries $X_c = \{u\in P : c\in u\}$. At the end of a $(3,n)$-selector, the $n-3$ lower lines carry $\emptyset$, and the three output lines carry $L_3\cup A_1$, $L_3\cup A_2$, $L_3\cup A_3$, where L_3 is the set of inputs of weight 3 and every input of weight 2 lies in exactly two of the sets A_r. Hence every $(3,n)$-selector has at least

$$\min_A \sum_{r=1}^{3} w(L_3\cup A_r) \;-\; n\,w(X_0)$$

comparators, the minimum taken over these arrangements. The largest such bound for this P, $\mathrm{LTP}(3,n)$, is the optimum of a linear program in the values of w and two more variables: maximize $z_{\text{out}} + z_{\text{low}} - n\,w(X_0)$ subject to the defect inequalities, $z_{\text{out}} \leq \sum_{r=1}^{3} w(L_3\cup A_r)$ for every arrangement A, and $z_{\text{low}} \leq (n-3)\,w(\emptyset) = 0$ for the lower lines.

Proposition 9.1. $\mathrm{LTP}(3,n) = 3, 6, 8$ *for* $n = 4,5,6$*, which equals* $\hat{U}_3(n)$*, and* $\mathrm{LTP}(3,7) \leq 10 < 11 = \hat{U}_3(7)$*.*

For $n=4,5,6$ the lower bounds are rational potentials, checked in exact arithmetic against every row of the program, and the upper bounds are exact dual solutions that use only rows coming from comparators of actual selectors and actual final arrangements. For $n=7$ an exact dual solution with ten rows proves the bound: eight defect rows taken from comparators of optimal $(3,7)$-selectors (one with multiplier 3), one row for a final arrangement, and one row for

the lower lines. Each potential and each dual solution is checked by two independent programs. By Proposition 9.1, lead-pattern potentials on this set P cannot prove $\hat{U}_3(7) = 11$, although potentials of this kind settle small merging problems [11]. We have not tested other sets P or other values of t; the full program for $n = 7$ is out of reach already on these two weights, whose up-sets number 75,268,594,037.

10. Formal verification

The selection results are also proved in Lean 4 [4], using only its core library; the development is archived at [12]. In the Lean development, a standard network selects the t largest if it moves the t largest of every input of natural numbers to the top t lines. The 0-1 reduction of Lemma 2.1 and the untangling of generalized networks are proved there. Table 4 gives the status of each result: proved by the kernel alone, with only the axioms `propext`, `Classical.choice` and `Quot.sound`; proved through compiled checks, which add auxiliary axioms and rely on the Lean compiler; or not formalized.

The upper bounds of Theorems 1.1 and 1.3 are checked on all 2^n inputs. The lower bounds of Theorems 1.1 and 1.2, and of Theorem 1.3 for $n \le 10$, come from the closed certificates of Section 6 and from Lemma 1.4. For $\hat{U}_3(n)$ with $n = 4, 5, 7$ and $\hat{U}_4(n)$ with $n = 6, 7$, the certificate stores, for each state and each move, the state that the move leads to and the permutation of the lines that relabels it, and the kernel checks each relabeling. For $\hat{U}_3(11)$ and $\hat{U}_4(n)$ with $n = 8, 9, 10$, compiled checks (`native_decide`) compute canonical forms instead: 22 calls for $\hat{U}_3(11)$ and 16, 19, 32 for $n = 8, 9, 10$, each adding one auxiliary axiom. In a test, a copy of a certificate with one stored permutation changed is rejected.

The bounds of Section 5 are in four modules. `LTP/Select/Alek2.lean` proves Theorem A as `select_lower_pow2`, its case $t = 8$ as `U8_lower_4n_sub_30`, and the bound of Remark 5.8 as `select_lower_classes`. `LTP/Select/Alek5.lean` proves Theorem B as `select_lower_pow2_band`, its case $t = 64$ as `U64_lower_7n_sub_441`, and the second bound of Corollary D for $t \ge 16$ as `select_lower_general_band`. `LTP/Select/Alek.lean` proves the case $t = 4$ as `U4_lower_3n_sub_11`, and the bound $\hat{U}_t(n) \ge (n - t)\lceil \lg(t + 1) \rceil + 1$ for $t \ge 3$ as `select_lower_alek_succ`. `LTP/Select/MaxStep.lean` proves the second bound of Corollary D for $t < 16$ as `select_lower_general`. In both proofs of that bound the pruning steps remove at least one comparator each, where Lemma 5.6 removes $\lceil \lg \lceil n/t \rceil \rceil$.

Open questions

Yao's question on $\hat{U}_3(n)$ is open from $n = 19$ on, where $\hat{U}_3(19)$ is 36 or 37. Knuth's expected equality $\hat{U}_4(n) = 3n - 10$ is open for every $n \ge 13$, where $\hat{U}_4(n)$ is $3n - 11$ or $3n - 10$. At powers of two $t \ge 8$ we do not know how far Theorems A–C are from $\hat{U}_t(n)$. In the weight model the least value of $\sigma + \varepsilon$ is known for $k \le 6$; for larger k we know only the lower bound $k - 1 + \mathrm{LB}(k)$ of Theorem C, and not how fast this least value grows.

Data availability

The supplement [13], doi:10.5281/zenodo.23801197, contains both search programs, with their logs, and the parallel driver used for $\hat{U}_4(12)$; Yao's networks and the optimal $(4, n)$-selectors, with independent checkers; the two programs for the restricted searches of Section 8, and the data of Table 3; the potentials and dual solutions of Section 9, with two checkers each; the checks of Section 5, among them a replay of its sequences, two computations of $\mathrm{LB}(k)$ and the searches in the weight model; and a separate check that the certificates of Section 6 are closed. Its README explains how to re-run each check. The Lean development of Section 10, with these certificates, is archived at [12], doi:10.5281/zenodo.23801200.

result	status in Lean
Lemma 1.4	kernel only
Theorem 1.1, $4 \le n \le 10$	kernel only
Theorem 1.1, $11 \le n \le 18$	compiled checks
Theorem 1.2	compiled checks
Theorem 1.3, $n = 6, 7$	kernel only
Theorem 1.3, $n = 8, 9, 10$	compiled checks
Theorem 1.3, $n = 11, 12$	the bound $3n - 11$ kernel only; the lower bound $3n - 10$ not formalized
Theorem A, with its cases $t = 4, 8$	kernel only
Theorem B, with its case $t = 64$	kernel only
Theorem C; Remark 5.4	not formalized
Proposition 5.5	not formalized
Corollary D, second bound	kernel only
Corollary D, first bound; Lemma 5.6	not formalized
Corollary 5.7	the bound $3n - 11$ kernel only; $\hat{U}_4(5) = 4$ and the upper bound for $n \ge 13$ not formalized
Remark 5.8	kernel only
Remark 4.2; Sections 8 and 9	not formalized

TABLE 4. The status of each result in the Lean development. The upper bounds of Theorems 1.1 and 1.3 are proved by the kernel alone for every n in their ranges; the compiled checks are those of the lower bounds. The single Lean statements of Theorem 1.1 and of $\hat{U}_4(n) = 3n - 10$ for $6 \le n \le 10$ use the compiled checks; the rows marked kernel only rest on the statements for single values of n and on Lemma 1.4.

REFERENCES

1. V. E. Alekseev, *Sorting algorithms with minimum memory*, Kibernetika **5** (1969), no. 5, 99–103, English translation: Cybernetics 5 (1969), 642–648.
2. Roberto Asín, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell, *Cardinality networks: a theoretical and empirical study*, Constraints **16** (2011), no. 2.
3. Michael Codish and Moshe Zazon-Ivry, *Pairwise cardinality networks*, Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16), Lecture Notes in Comput. Sci., vol. 6355, Springer, 2010, pp. 154–172.
4. Leonardo de Moura and Sebastian Ullrich, *The Lean 4 theorem prover and programming language*, Automated Deduction – CADE 28, Lecture Notes in Comput. Sci., vol. 12699, Springer, 2021, pp. 625–635.
5. Shuji Jimbo and Akira Maruoka, *A method of constructing selection networks with $O(\log n)$ depth*, SIAM J. Comput. **25** (1996), no. 4.
6. Michał Karpiński and Marek Piotrów, *Smaller selection networks for cardinality constraints encoding*, Principles and Practice of Constraint Programming (CP 2015), Lecture Notes in Comput. Sci., Springer, 2015, arXiv:1502.04551.
7. Donald E. Knuth, *Errata for The Art of Computer Programming, vol. 3, second edition*, `https://www-cs-faculty.stanford.edu/~knuth/taocp.html`, Entries of 12 September 2003: replacement of exercise 5.3.4–20 and its answer.
8. ———, *The art of computer programming, vol. 3: Sorting and searching*, 2nd ed., Addison-Wesley, Reading, MA, 1998.
9. Nicholas Pippenger, *Selection networks*, SIAM J. Comput. **20** (1991), no. 5, 878–887.
10. Yicheng Qu, Amir Tasbihi, and Frank R. Kschischang, *Constituent automorphism decoding of Reed–Muller codes*, IEEE Trans. Commun. (2025), arXiv:2409.03700.
11. Tejasvi Singh Tomar, *Lead-pattern potentials and exact values of merging networks*, Zenodo, version 5, doi:10.5281/zenodo.23756850, 2026.
12. ———, *Lean development for "Exact sizes of networks that select the three or four largest of n elements"*, Zenodo, doi:10.5281/zenodo.23801200, 2026.

13. ———, *Supplement to "Exact sizes of networks that select the three or four largest of n elements"*, Zenodo, doi:10.5281/zenodo.23801197, 2026, Checksum file `SHA256SUMS` with SHA-256 hash `26f96dbce112bfad`....
14. Andrew C. Yao, *Bounds on selection networks*, 15th Annual Symposium on Switching and Automata Theory, 1974, pp. 110–116.
15. ———, *Bounds on selection networks*, SIAM J. Comput. **9** (1980), no. 3, 566–582.
16. Andrew Chi-Chih Yao, *A study of concrete computational complexity*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1975, Report UIUCDCS-R-75-716.

ORCID: `https://orcid.org/0000-0003-4668-3639`