261/2004

Raport Badawczy

Research Report

RB/67/2004

Improved Randomized Selection

Krzysztof C. Kiwiel

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Krzysztof C. Kiwiel

Warszawa 2004

Improved randomized selection

Krzysztof C. Kiwiel*

February 2, 2004

Abstract

We show that several versions of Floyd and Rivest's improved algorithm SELECT for finding the kth smallest of n elements require at most $n + \min\{k, n-k\} + O(n^{1/2} \ln^{1/2} n)$ comparisons on average and with high probability. This rectifies the analysis of Floyd and Rivest, and extends it to the case of nondistinct elements. Encouraging computational results on large median-finding problems are reported.

Key words. Selection, medians, computational complexity.

1 Introduction

The *selection problem* is defined as follows: Given a set $X := \{x_j\}_{j=1}^n$ of n elements, a total order $<$ on X, and an integer $1 \le k \le n$, find the *kth smallest* element of X, i.e., an element x of X for which there are at most $k-1$ elements $x_j < x$ and at least k elements $x_j \le x$. The *median* of X is the $\lceil n/2 \rceil$th smallest element of X.

Selection is one of the fundamental problems in computer science; see, e.g., the references in [DHUZ01, DoZ99, DoZ01] and [Knu98, §5.3.3]. Most references concentrate on the number of comparisons between pairs of elements made in selection algorithms. In the worst case, selection needs at least $(2+\epsilon)n$ comparisons [DoZ01], whereas the algorithm of [BFP+72] makes at most $5.43n$, that of [SPP76] needs $3n + o(n)$, and that in [DoZ99] takes $2.95n + o(n)$. In the average case, for $k \le \lceil n/2 \rceil$, at least $n + k - O(1)$ comparisons are necessary [CuM89], whereas Knuth's *best upper bound* is $n + k + O(n^{1/2} \ln^{1/2} n)$ [Knu98, Eq. (5.3.3.16)]. The classical algorithm FIND of [Hoa61], also known as quickselect, has an upper bound of $3.39n + o(n)$ for $k = \lceil n/2 \rceil$ in the average case [Knu98, Ex. 5.2.2–32], which improves to $2.75n + o(n)$ for median-of-3 pivots [Grü99, KMP97].

The seminal papers [FlR75a, FlR75b] presented three versions of the algorithm SELECT with very good average case performance, although their analysis had gaps, as noted in [PRKT83] and [Knu98, Ex. 5.3.3–24]. Our recent papers [Kiw03b, Kiw04] rectified the analysis of [FlR75b, §2.2] and extended it to the case of nondistinct elements. Specifically, we showed that several versions of SELECT, close to those in [FlR75b, §2.1] and [FlR75a], make at most $n + k + O(n^{2/3} \ln^{1/3} n)$ comparisons on average.

*Systems Research Institute, Newelska 6, 01–447 Warsaw, Poland (kiwiel@ibspan.waw.pl)

This paper concentrates on versions of the improved SELECT from [FlR75b, §2.3], again correcting its analysis and extending it to the case of nondistinct elements. We show that they make at most $n + k + O(n^{1/2} \ln^{1/2} n)$ comparisons on average.

Thus, apparently for the first time, Knuth's best upper bound is attained by an *implementable* algorithm without *restrictive* assumptions. Specifically, Knuth's scheme [Knu98, Ex. 5.3.3–24] is not formulated precisely enough to qualify as an algorithm, it requires distinct elements in random order, and its samples are too large for efficient randomization (since generating a random sample of size $\lceil n/2 \rceil$ takes too much time; cf. §6.3).

We also prove that nonrecursive versions of SELECT, which employ other linear-time selection routines for small subproblems, require at most $n+k+O(n^{1/2} \ln^{1/2} n)$ comparisons with high probability; we couldn't find such results in the literature. When sorting routines are used, the bound becomes $n + k + O(n^{1/2} \ln^{3/2} n)$.

Since our interest is not merely theoretical, a serious effort was made to implement the various versions efficiently and to test them in practice. Our tests on the median-finding examples of [Val00] show that the improved SELECT is as fast as the ternary version of [Kiw04], although a bit slower than the quintary version of [Kiw03b]. All these versions perform very well in terms of the number of comparisons made on large inputs, the average numbers being about $1.6n$ for $n = 1$M, and as small as $1.53n$ for $n = 16$M. Since the lower bound is $1.5n$, little room for improvement remains. Of course, future work should assess more fully the relative merits of these versions, but clearly the improved SELECT may compete with other methods in both theory and practice.

The paper is organized as follows. A simplified version of SELECT that ignores some roundings is introduced in §2, and its basic features are analyzed in §3. The average performance of SELECT and its practical rounded versions is studied in §4. High probability bounds for nonrecursive versions are derived in §5. Finally, our computational results are reported in §6.

Our notation is fairly standard. $|A|$ denotes the cardinality of a set A. In a given probability space, P is the probability measure, and E is the mean-value operator.

2 The algorithm SELECT

We first recall that the standard version of SELECT proceeds as follows. By solving two pivot selection subproblems over a random sample S from X, two elements u and v almost sure to be just below and above the kth are found. The remaining elements are compared with u and v to derive a reduced selection problem on the elements between u and v that is solved recursively. In general, the size of the reduced problem (and hence its cost) diminishes when a larger sample is used, but then the cost of pivot selection grows. To balance these costs, the standard version employs a relatively small sample. In contrast, the improved version uses a much larger "final" sample S, but u and v are selected iteratively by using samples from S. More specifically, let $S_1 \subset \cdots \subset S_{\bar{l}} \subset S_{\bar{l}+1} = X$ be a nested series of random samples from X. For each sample S_l, two pivots u_l and v_l are found such that $u_l \le x_k^* \le v_l$ with high probability, where x_k^* is the kth element of X. In particular, $u_l = x_k^* = v_l$ when $S_l = X$. For $l \le \bar{l}$, the positions of u_{l+1} and v_{l+1} in S_{l+1} are chosen so that $u_l \le u_{l+1} \le v_{l+1} \le v_l$ with high probability, and hence u_l and v_l can be used to bound the search for u_{l+1} and v_{l+1}.

For clarity, we first describe SELECT in detail without some integer round-ups in sample sizes, etc.; more practical versions are postponed till §4.2.

Algorithm 2.1.
SELECT(X, k) (Selects the kth smallest element of X, with $1 \le k \le n := |X|$)

Step 1 (*Initiation*). If $n = 1$, return x_1. Choose parameters $\alpha \in (0, 1/2]$, $s_1 := n^\alpha$, $r > 1$, $\kappa := 1/r$, $\beta \ge \frac{1}{4}(1-\kappa)^{-2}$, and $\bar{l}$ such that $n = r^{2\bar{l}} s_1$. Set $\theta := k/n$ and $l := 1$.

Step 2 (*Initial sample selection*). Draw a random sample S_1 of size s_1 from X. Set

$$g_l := \begin{cases} (\beta s_l \ln n)^{1/2} & \text{if } l \le \bar{l}, \\ 0 & \text{if } l = \bar{l} + 1, \end{cases} \tag{2.1}$$

$$i_u^l := \max\{\lceil \theta s_l - g_l \rceil, 1\} \quad \text{and} \quad i_v^l := \min\{\lceil \theta s_l + g_l \rceil, s_l\}, \tag{2.2}$$

$u_1 :=$ SELECT(S_1, i_u^1) and $v_1 :=$ SELECT(S_1, i_v^1) by using SELECT recursively.

Step 3 (*Sample selection*). Draw a random sample S_{l+1} of size $s_{l+1} := r^2 s_l$ from X such that $S_l \subset S_{l+1}$. (Here $s_{l+1} - s_l$ elements of $X \setminus S_l$ are picked randomly.)

Step 4 (*Partitioning*). By comparing each element x of $S_{l+1} \setminus S_l$ to $u := u_l$ and $v := v_l$, partition S_{l+1} into $L := \{x \in S_{l+1} : x < u\}$, $U := \{x \in S_{l+1} : x = u\}$, $M := \{x \in S_{l+1} : u < x < v\}$, $V := \{x \in S_{l+1} : x = v\}$, $R := \{x \in S_{l+1} : v < x\}$. If $\theta < 1/2$, x is compared to v first, and to u only if $x < v$. If $\theta \ge 1/2$, the order of the comparisons is reversed.

Step 5 (*Pivot selection*). (a) Set g_{l+1}, $i_u^+ := i_u^{l+1}$ and $i_v^+ := i_v^{l+1}$ via (2.1)–(2.2). (Here we wish to find u_{l+1} and v_{l+1} as the i_u^+th and i_v^+th smallest elements of S_{l+1}.)
(b) If $|L| < i_u^+ \le |L \cup U|$, set $u_{l+1} := u$; else if $|L \cup U \cup M| < i_u^+ \le s_{l+1} - |R|$, set $u_{l+1} := v$; else set $u_{l+1} :=$ SELECT$(\hat{S}_u, \hat{\imath}_u^+)$, where $\hat{S}_u$ and $\hat{\imath}_u^+$ are determined as follows. If $i_u^+ \le |L|$, set $\hat{S}_u := L$ and $\hat{\imath}_u^+ := i_u^+$; else if $s_{l+1} - |R| < i_u^+$, set $\hat{S}_u := R$ and $\hat{\imath}_u^+ := i_u^+ - s_{l+1} + |R|$; else set $\hat{S}_u := M$ and $\hat{\imath}_u^+ := i_u^+ - |L \cup U|$.
(c) Find v_{l+1}, and possibly $\hat{S}_v$ and $\hat{\imath}_v^+$, as in (b) with i_u^+ replaced by i_v^+ and u_{l+1} by v_{l+1}.

Step 6 (*Loop*). If $s_{l+1} = n$, return u_{l+1}. Otherwise, increase l by 1 and go to Step 3.

A few remarks on the algorithm are in order.

Remarks 2.2. (a) The correctness and finiteness of SELECT stem by induction from the following observations. At Step 2, $|S_1| < |X|$. At Step 5, $\hat{S}_u$ and $\hat{\imath}_u^+$ are chosen so that the i_u^+th smallest element of S_{l+1} is the $\hat{\imath}_u^+$th smallest element of $\hat{S}_u$, and $|\hat{S}_u| < s_{l+1}$ (since $u, v \notin \hat{S}_u$); similarly for $\hat{S}_v$ and $\hat{\imath}_v^+$. The final loop with $l = \bar{l}$ has $S_{l+1} = X$, $g_{l+1} = 0$ and $i_u^+ = \theta n = k$, so $u_{l+1} = v_{l+1}$ is the desired element.

(b) After Step 5 the position of each element of S_{l+1} relative to u_{l+1} and v_{l+1} is known. Hence Step 4 need only compare u and v with the elements of $S_{l+1} \setminus S_l$ (e.g., via one of the quintary partitioning schemes of [Kiw03b, §6]).

(c) The following elementary property is needed in §4.1. The maximum number of comparisons taken by SELECT on any input of size n is finite, for each n (because the recursive calls of Steps 2 and 5 deal with proper subsets of X).

3 Preliminary analysis

In this section we analyze general features of sampling used by SELECT.

3.1 Sampling deviations and expectation bounds

Our analysis hinges on the following bound on the tail of the hypergeometric distribution established in [Hoe63] and rederived shortly in [Chv79].

Fact 3.1. *Let s balls be chosen uniformly at random from a set of s_+ balls, of which ρ are red, and ρ' be the random variable representing the number of red balls drawn. Let $p := \rho/s_+$. Then*

$$\mathrm{P}[\rho' \geq ps + g] \leq e^{-2g^2/s} \quad \forall g \geq 0. \tag{3.1}$$

We shall also need a simple version of the (left) Chebyshev inequality [Kor78, §2.4.2].

Fact 3.2. *Let η be a nonnegative random variable such that $\mathrm{P}[\eta \leq \zeta] = 1$ for some constant ζ. Then $\mathrm{E}\eta \leq t + \zeta \mathrm{P}[\eta > t]$ for all nonnegative real numbers t.*

3.2 Sample ranks and partitioning efficiency

In this subsection we analyze in detail a fixed iteration l of SELECT.

For simpler notation, we drop l from the subscripts and superscripts and replace $l+1$ by $+$. Thus let $y_1^* \leq \dots y_s^*$ and $z_1^* \leq \dots z_{s_+}^*$ denote the sorted elements of the samples S and S_+, so that $u = y_{i_u}^*$, $v = y_{i_v}^*$, $u_+ = z_{i_u^+}^*$ and $v_+ = z_{i_v^+}^*$, where

$$i_u := \max\{\lceil \theta s - g \rceil, 1\} \quad \text{and} \quad i_v := \min\{\lceil \theta s + g \rceil, s\}, \tag{3.2}$$

$$i_u^+ := \max\{\lceil \theta s_+ - g_+ \rceil, 1\} \quad \text{and} \quad i_v^+ := \min\{\lceil \theta s_+ + g_+ \rceil, s_+\}. \tag{3.3}$$

This notation facilitates showing that $u \leq u_+ \leq v_+ \leq v$ with high probability. To deduce that the number of elements between u and v is small enough, let

$$j_u := \max\{\lceil \theta s_+ - 2gs_+/s \rceil, 1\} \quad \text{and} \quad j_v := \min\{\lceil \theta s_+ + 2gs_+/s \rceil, s_+\} \tag{3.4}$$

be bounding indices; we shall see that $z_{j_u}^* \leq u \leq v \leq z_{j_v}^*$ with high probability. Our argument is similar to that of [Kiw03b, Lem. 3.3] because S may be regarded as a random sample from S_+; the key difference is that $g_+ \neq 0$ in (3.3) if $l < \bar{l}$, in which case g is replaced by $(1-\kappa)g$ in our probability bounds. To this end, note that, since $\kappa := 1/r = (s/s_+)^{1/2}$, (2.1) yields

$$g - g_+ s/s_+ = \begin{cases} (1-\kappa)g & \text{if } l < \bar{l}, \\ g & \text{otherwise.} \end{cases} \tag{3.5}$$

Lemma 3.3. (a) $\mathrm{P}[u_+ < u] \leq e^{-2(1-\kappa)^2 g^2/s}$ *if* $i_u = \lceil \theta s - g \rceil$.

(b) $\mathrm{P}[u < z_{j_u}^*] \leq e^{-2g^2/s}$.

(c) $\mathrm{P}[v < v_+] \leq e^{-2(1-\kappa)^2 g^2/s}$ *if* $i_v = \lceil \theta s + g \rceil$.

(d) $\mathrm{P}[z_{j_v}^* < v] \leq e^{-2g^2/s}$.

(e) $i_u \neq \lceil \theta s - g \rceil$ *iff* $\theta \leq g/s$; $i_v \neq \lceil \theta s + g \rceil$ *iff* $1 < \theta + g/s$.

Proof. (a) If $z^*_{i^+_u} < y^*_{i_u}$, at least $s - i_u + 1$ samples satisfy $y_i \geq z^*_{\bar{\jmath}+1}$ with $\bar{\jmath} := \max_{z^*_j = z^*_{i^+_u}} j$. In the setting of Fact 3.1, we have $\rho := s_+ - \bar{\jmath}$ red elements $z_j \geq z^*_{\bar{\jmath}+1}$, $ps = s - \bar{\jmath}s/s_+$ and $\rho' \geq s - i_u + 1$. Since $i_u = \lceil \theta s - g \rceil < \theta s - g + 1$ and $\bar{\jmath} \geq i^+_u \geq \theta s_+ - g_+$ by (3.3), we get $s - i_u + 1 - ps > \bar{\jmath}s/s_+ - \theta s + g \geq g - g_+ s/s_+$; thus $\rho' \geq ps + (1-\kappa)g$ by (3.5). Hence $\mathrm{P}[u_+ < u] \leq \mathrm{P}[\rho' \geq ps + (1-\kappa)g]$, and (3.1) yields the conclusion.

(b) If $y^*_{i_u} < z^*_{j_u}$, i_u samples are at most z^*_ρ, where $\rho := \max_{z^*_j < z^*_{j_u}} j$. Thus we have ρ red elements $z_j \leq z^*_\rho$, $ps = \rho s/s_+$ and $\rho' \geq i_u$. Now, $1 \leq \rho \leq j_u - 1$ implies $2 \leq j_u = \lceil \theta s_+ - 2gs_+/s \rceil$ by (3.4) and thus $j_u < \theta s_+ - 2gs_+/s + 1$, so $-\rho s/s_+ > -\theta s + 2g$. Hence $i_u - ps - g \geq \theta s - g - \rho s/s_+ - g > 0$, i.e., $\rho' > ps + g$; invoke (3.1) as before.

(c) and (d): Argue symmetrically to (a) and (b); cf. [Kiw03b, Proof of Lem. 3.3].

(e) Follows immediately from the properties of $\lceil \cdot \rceil$ [Knu97, §1.2.4]. □

We may now estimate the partitioning costs of Step 4.

Lemma 3.4. *Let $c := c_l$ denote the number of comparisons made at Step 4. Then*

$$\mathrm{P}[c < \bar{c}] \geq 1 - e^{-2g^2/s} \quad \textit{and} \quad \mathrm{E}c \leq \bar{c} + 2(s_+ - s)e^{-2g^2/s} \quad \textit{with} \tag{3.6a}$$

$$\bar{c} := (1 + \min\{\theta, 1-\theta\})(s_+ - s) + 3gs_+/s. \tag{3.6b}$$

Proof. Consider the event $\mathcal{A} := \{c < \bar{c}\}$ and its complement $\mathcal{A}' := \{c \geq \bar{c}\}$. If $u = v$ then $c = s_+ - s < \bar{c}$; hence $\mathrm{P}[\mathcal{A}'] = \mathrm{P}[\mathcal{A}' \cap \{u < v\}]$, and we may assume $u < v$ below.

First, suppose $\theta < 1/2$. Then $c = s_+ - s + |\{z \in S_+ \setminus S : z < v\}|$, since $s_+ - s$ elements of $S_+ \setminus S$ are compared to v first. In particular, $c \leq 2(s_+ - s)$. If $v \leq z^*_{j_v}$, then $\{z \in S_+ : z < v\} \subset \{z \in S_+ : z < z^*_{j_v}\}$ gives $|\{z \in S_+ : z < v\}| \leq j_v - 1 < \theta s_+ + 2gs_+/s$ by (3.4), whereas $u < v$ implies $|\{z \in S : z < v\}| \geq |\{z \in S : z \leq u\}| \geq i_u \geq \theta s - g$ by (3.2), so $|\{z \in S_+ \setminus S : z < v\}| < \theta(s_+ - s) + 2gs_+/s + g$ yields $c < \bar{c}$. Thus $u < v \leq z^*_{j_v}$ implies $\mathcal{A}$. Therefore, $\mathcal{A}' \cap \{u < v\}$ implies $\{z^*_{j_v} < v\} \cap \{u < v\}$, so $\mathrm{P}[\mathcal{A}' \cap \{u < v\}] \leq \mathrm{P}[z^*_{j_v} < v] \leq e^{-2g^2/s}$ (Lem. 3.3(d)). Hence we have (3.6), since $\mathrm{E}c \leq \bar{c} + 2(s_+ - s)e^{-2g^2/s}$ by Fact 3.2 (with $\eta := c$, $\zeta := 2(s_+ - s)$).

Next, suppose $\theta \geq 1/2$. Now $c = s_+ - s + |\{z \in S_+ \setminus S : u < z\}|$, since $s_+ - s$ elements of $S_+ \setminus S$ are compared to u first. If $z^*_{j_u} \leq u$, then $\{z \in S_+ : u < z\} \subset \{z \in S_+ : z^*_{j_u} < z\}$ gives $|\{z \in S_+ : u < z\}| \leq s_+ - j_u \leq s_+ - \theta s_+ + 2gs_+/s$, whereas $u < v$ implies $|\{z \in S : u < z\}| \geq |\{z \in S : v \leq z\}| \geq s - i_v + 1 \geq s - \theta s - g + 1$, so $|\{z \in S_+ \setminus S : u < z\}| \leq (1-\theta)(s_+ - s) + 2gs_+/s + g - 1$ yields $c < \bar{c}$. Thus $\mathcal{A}' \cap \{u < v\}$ implies $\{u < z^*_{j_u}\} \cap \{u < v\}$, so $\mathrm{P}[\mathcal{A}' \cap \{u < v\}] \leq \mathrm{P}[u < z^*_{j_u}] \leq e^{-2g^2/s}$ (Lem. 3.3(b)), and we get (3.6) as before. □

The following result will imply that the sets $\hat{S}_u$ and $\hat{S}_v$ selected at Step 5 are "small enough" with high probability. Let $\hat{s} := \hat{s}_l := |\hat{S}_u \cup \hat{S}_v|$; we let $\hat{S}_u := \emptyset$ (or $\hat{S}_v := \emptyset$) if Step 5 doesn't use $\hat{S}_u$ (or $\hat{S}_v$), but we don't consider this case explicitly.

Lemma 3.5. $\mathrm{P}[\hat{s} < 4gs_+/s] \geq 1 - \mathrm{P}_{\mathsf{fail}}$ *and $\hat{s} < s_+$ always, where*

$$\mathrm{P}_{\mathsf{fail}} := \mathrm{P}_{\mathsf{fail}}(n) := 2e^{-2g^2/s} + 2e^{-2(1-\kappa)^2 g^2/s} = 2n^{-2\beta} + 2n^{-2(1-\kappa)^2\beta} \leq 4n^{-2(1-\kappa)^2\beta}. \tag{3.7}$$

Proof. First, consider the *middle* case of $i_u = \lceil \theta s - g \rceil$ and $i_v = \lceil \theta s + g \rceil$. Let $\mathcal{E}$ denote the event $z^*_{j_u} \le u \le u_+ \le v_+ \le v \le z^*_{j_v}$. By Lem. 3.3 and the Boole-Benferroni inequality, its complement $\mathcal{E}'$ has $\mathrm{P}[\mathcal{E}'] \le \mathrm{P}_{\mathsf{fail}}$, so $\mathrm{P}[\mathcal{E}] \ge 1 - \mathrm{P}_{\mathsf{fail}}$. By the rules of Steps 4–5, $u \le u_+ \le v_+ \le v$ implies $\hat{S}_u \cup \hat{S}_v \subset M$, whereas $z^*_{j_u} \le u \le v \le z^*_{j_v}$ yields $\hat{s} \le j_v - j_u + 1 - 2$; since $j_v < \theta s_+ + 2gs_+/s + 1$ and $j_u \ge \theta s_+ - 2gs_+/s$ by (3.4), we get $\hat{s} < 4gs_+/s$. Hence $\mathrm{P}[\hat{s} < 4gs_+/s] \ge \mathrm{P}[\mathcal{E}]$. Then (3.7) follows from (2.1) and the fact $\kappa \in (0, 1)$.

Next, consider the *left* case of $i_u \ne \lceil \theta s - g \rceil$, i.e., $\theta \le g/s$ (Lem. 3.3(e)). If $i_v \ne \lceil \theta s + g \rceil$, then $1 < \theta + g/s$ (Lem. 3.3(e)) gives $\hat{s} < s_+ < 2gs_+/s$. For $i_v = \lceil \theta s + g \rceil$, $\mathrm{P}[v_+ \le v \le z^*_{j_v}] \ge 1 - \frac{1}{2}\mathrm{P}_{\mathsf{fail}}$ by Lem. 3.3(c,d). Now, $v_+ \le v$ implies $\hat{S}_u \cup \hat{S}_v \subset L \cup M$, whereas $v \le z^*_{j_v}$ gives $\hat{s} \le j_v - 1 < \theta s_+ + 2gs_+/s \le 3gs_+/s$; hence $\mathrm{P}[\hat{s} < 4gs_+/s] \ge \mathrm{P}[v_+ \le v \le z^*_{j_v}]$.

Finally, consider the *right* case of $i_v \ne \lceil \theta s + g \rceil$, i.e., $1 < \theta + g/s$. If $i_u \ne \lceil \theta s - g \rceil$ then $\theta \le g/s$ gives $\hat{s} < s_+ < 2gs_+/s$. For $i_u = \lceil \theta s - g \rceil$, we have $\mathrm{P}[z^*_{j_u} \le u \le u_+] \ge 1 - \frac{1}{2}\mathrm{P}_{\mathsf{fail}}$ by Lem. 3.3(a,b). Now, $u \le u_+$ implies $\hat{S}_u \cup \hat{S}_v \subset M \cup R$, whereas $z^*_{j_u} \le u$ yields $\hat{s} \le s_+ - j_u$ with $j_u \ge \theta s_+ - 2gs_+/s$ and thus $\hat{s} < 3gs_+/s$, so $\mathrm{P}[\hat{s} < 4gs_+/s] \ge \mathrm{P}[z^*_{j_u} \le u \le u_+]$. □

Corollary 3.6. $\mathrm{P}\,[c < \bar{c}$ *and* $\hat{s} < 4gs_+/s] \ge 1 - \mathrm{P}_{\mathsf{fail}}$.

Proof. If $2g/s \ge 1$ then $c \le 2(s_+ - s) < \bar{c}$ (cf. (3.6b)) and $\hat{s} < s_+ < 4gs_+/s$, so assume $2g/s < 1$. The conclusion follows from the proofs of Lems. 3.4 and 3.5. We only note that the left case of $\theta \le g/s$ now has $i_v = \lceil \theta s + g \rceil$ and $\theta < 1/2$. Similarly, in the right case of $1 < \theta + g/s$, we have $i_u = \lceil \theta s - g \rceil$ and $\theta \ge 1/2$, since $g/s < 1/2$. □

Remark 3.7. Suppose for $l < \bar{l}$, Step 5 resets $i^+_u := i^+_v$ if $\theta \le g_{l+1}/s_{l+1}$, or $i^+_v := i^+_u$ if $1 < \theta + g_{l+1}/s_{l+1}$, finding a single pivot $u_+ = v_+$ in these cases. The preceding results remain valid for this modification (which corresponds to using $u := v$ if $\theta \le g/s$, or $v := u$ if $1 < \theta + g/s$). Similarly, Step 2 may reset $i^1_u := i^1_v$ if $\theta \le g_1/s_1$, or $i^1_v := i^1_u$ if $1 < \theta + g_1/s_1$.

4 Average performance of the recursive version

4.1 Analysis of the nonrounded version

In this section we analyze the average performance of SELECT, starting with the "nonrounded" version of Algorithm 2.1; more practical versions are discussed in §4.2.

Theorem 4.1. *Let C_{nk} denote the expected number of comparisons made by* SELECT*, and $f(t) := (t \ln t)^{1/2}$ for $t \ge 1$. There exists a positive constant γ such that*

$$C_{nk} \le n + \min\{k, n-k\} + \gamma f(n) \quad \textit{for all } 1 \le k \le n. \tag{4.1}$$

Proof. We need a few preliminary facts. The function $\phi(t) := f(t)/t = (\ln t/t)^{1/2}$ decreases to 0 on $[e, \infty)$, whereas $f(t)$ grows to infinity on $[2, \infty)$. The key *bounding property* is $f(t) = \phi(t)t \le \phi(\hat{t})t$ for all $t \ge \hat{t} \ge e$. Pick $\bar{n} \ge 2$ large enough so that $s_1 \ge e$, $4r^2 g_1 \ge e$, $n^\alpha + 1 \le f(n)$ and $n \le r^2 s_1$ for all $n \ge \bar{n}$. Using $\alpha \in (0, 1/2]$ and the bounding property, we have

$$s_1 \le f(n) \quad \text{and} \quad f(s_1) \le \phi(s_1) f(n). \tag{4.2}$$

By (3.7) and our assumption $\beta \geq \frac{1}{4}(1-\kappa)^{-2}$, we have $n\mathrm{P_{fail}}(n) = o(f(n))$; more precisely,

$$n\mathrm{P_{fail}}(n) \leq 4n^{1-2(1-\kappa)^2\beta} = 4f(n)n^{1/2-2(1-\kappa)^2\beta}\ln^{-1/2} n. \tag{4.3}$$

Using the monotonicity of ϕ, we may increase $\bar{n}$ if necessary to get for all $n \geq \bar{n}$

$$\phi(s_1) + 4\frac{2r^2-r}{r-1}\beta^{1/2}\phi(4r^2 g_1) + 4\frac{2r^2-1}{r^2-1}\phi(r^2 s_1)n^{1/2-2(1-\kappa)^2\beta}\ln^{-1/2} n \leq 0.475, \tag{4.4}$$

since each term above goes to 0 as n increases to ∞. By Rem. 2.2(c), there is γ such that (4.1) holds for all $n \leq \bar{n}$; increasing γ if necessary, we have for all $n \geq \bar{n}$

$$3 + 15\frac{2r^2-r}{r-1}\beta^{1/2} + 4\frac{6.5r^2-3.5}{r^2-1}n^{1/2-2(1-\kappa)^2\beta}\ln^{-1/2} n \leq 0.05\gamma. \tag{4.5}$$

Let $n' \geq \bar{n}$. Assuming (4.1) holds for all $n \leq n'$, for induction let $n = n'+1$.

Since $s_1 < n$, by our hypothesis the cost of selecting u_1 and v_1 at Step 2 is at most

$$C_{s_1 i_u^1} + C_{s_1 i_v^1} \leq 3s_1 + 2\gamma f(s_1). \tag{4.6}$$

Similarly, the cost of selecting u_{l+1} and v_{l+1} at Step 5 is at most $3\hat{s}_l + 2\gamma f(\hat{s}_l)$, where $\hat{s}_l < s_{l+1}$ and $\mathrm{P}[\hat{s}_l \geq 4g_l s_{l+1}/s_l] \leq \mathrm{P_{fail}}$ by Lem. 3.5. Hence (cf. Fact 3.2 with $\eta := 3\hat{s}_l + 2\gamma f(\hat{s}_l)$)

$$\mathrm{E}\left[3\hat{s}_l + 2\gamma f(\hat{s}_l)\right] \leq 12g_l s_{l+1}/s_l + 2\gamma f(4g_l s_{l+1}/s_l) + \left[3s_{l+1} + 2\gamma f(s_{l+1})\right]\mathrm{P_{fail}}, \quad l = 1\colon \bar{l}. \tag{4.7}$$

For $\bar{\theta} := \min\{\theta, 1-\theta\}$, the partitioning cost of Step 4 is estimated by (3.6) as

$$\mathrm{E}c_l \leq \left(1+\bar{\theta}\right)(s_{l+1}-s_l) + 3g_l s_{l+1}/s_l + \tfrac{1}{2}(s_{l+1}-s_l)\mathrm{P_{fail}}, \quad l = 1\colon \bar{l}. \tag{4.8}$$

Adding the costs (4.6)–(4.8) and using $s_{\bar{l}+1} = n$, we get

$$C_{nk} \leq \left(1+\bar{\theta}\right)(n-s_1) + \left[3s_1 + 15\sum_{l=1}^{\bar{l}} g_l s_{l+1}/s_l + \tfrac{1}{2}\mathrm{P_{fail}}(n-s_1) + 3\mathrm{P_{fail}}\sum_{l=1}^{\bar{l}} s_{l+1}\right] \tag{4.9a}$$

$$+ 2\gamma\left[f(s_1) + \sum_{l=1}^{\bar{l}} f(4g_l s_{l+1}/s_l) + \mathrm{P_{fail}}\sum_{l=1}^{\bar{l}} f(s_{l+1})\right]. \tag{4.9b}$$

Since $\theta := k/n$, the first term on the right side above is at most $n + \min\{k, n-k\}$. Next, for $d := (\beta \ln n)^{1/2}$, (2.1) yields $g_l s_{l+1}/s_l = d s_{l+1}/s_l^{1/2}$ for $l \leq \bar{l}$. Since $s_l = r^{2(l-1)}s_1$ for $l \leq \bar{l}$, and $n > r^{2(\bar{l}-1)}s_1$ implies $r^{\bar{l}-1} < (n/s_1)^{1/2}$, we obtain

$$\sum_{l=1}^{\bar{l}-1} g_l s_{l+1}/s_l = \sum_{l=1}^{\bar{l}-1} d r^{l+1} s_1^{1/2} = d r^2 s_1^{1/2}\frac{r^{\bar{l}-1}-1}{r-1} < \beta^{1/2} f(n)\frac{r^2}{r-1}.$$

But $g_{\bar{l}} s_{\bar{l}+1}/s_{\bar{l}} = dn/s_{\bar{l}}^{1/2} = \beta^{1/2} f(n)(n/s_{\bar{l}})^{1/2}$ and $n \leq r^2 s_{\bar{l}}$ imply $g_{\bar{l}} s_{\bar{l}+1}/s_{\bar{l}} \leq \beta^{1/2} f(n) r$, so

$$\sum_{l=1}^{\bar{l}} g_l s_{l+1}/s_l < \beta^{1/2} f(n)\left(\frac{r^2}{r-1} + r\right) = \beta^{1/2} f(n)\frac{2r^2-r}{r-1}. \tag{4.10}$$

Similarly, using $s_{l+1} = r^{2l}s_1$ for $l < \bar{l}$, $s_{\bar{l}+1} = n$ and $r^{2(\bar{l}-1)} < n/s_1$, we get

$$\sum_{l=1}^{\bar{l}} s_{l+1} = s_1 \sum_{l=1}^{\bar{l}-1} r^{2l} + n = r^2 s_1 \frac{r^{2(\bar{l}-1)}-1}{r^2-1} + n < r^2 \frac{n-s_1}{r^2-1} + n < \frac{2r^2-1}{r^2-1} n. \tag{4.11}$$

Plugging (4.2), (4.3), (4.10) and (4.11) into (4.9a), we see that the bracketed term is at most $0.05\gamma f(n)$ thanks to (4.5). Next, for $l < \bar{l}$ we have $4g_l s_{l+1}/s_l \geq 4r^2 g_1$ (cf. (2.1)), whereas $g_{\bar{l}} s_{\bar{l}+1}/s_{\bar{l}} \leq \beta^{1/2} f(n) r$ with $4\beta^{1/2} f(n) r \geq 4r^2 g_1$ from $n \geq r^2 s_1$; therefore, we may use the bounding property and argue as for (4.10) to get

$$\sum_{l=1}^{\bar{l}} f(4g_l s_{l+1}/s_l) \leq \phi(4r^2 g_1) 4 \left(\sum_{l=1}^{\bar{l}-1} g_l s_{l+1}/s_l + \beta^{1/2} f(n) r \right) < 4 \frac{2r^2 - r}{r-1} \beta^{1/2} \phi(4r^2 g_1) f(n). \tag{4.12}$$

Similarly, $s_{l+1} = r^{2l}s_1 \geq r^2 s_1$ for $l < \bar{l}$ and $s_{\bar{l}+1} = n \geq r^2 s_1$ together with (4.11) imply

$$\sum_{l=1}^{\bar{l}} f(s_{l+1}) \leq \phi(r^2 s_1) \sum_{l=1}^{\bar{l}} s_{l+1} < \frac{2r^2-1}{r^2-1} \phi(r^2 s_1) n. \tag{4.13}$$

Now, plugging (4.2), (4.12) and (4.13) combined with (4.3) into (4.9b), we deduce that (4.9b) is at most $0.95\gamma f(n)$ due to (4.4); thus (4.1) holds as required. □

4.2 Analysis of rounded versions

We now consider more realistic parameter choices for SELECT.

Fixing $\alpha \in (0, 1/2]$, $r > 1$ such that r^2 is integer, $\kappa := 1/r$, $\beta \geq \frac{1}{4}(1-\kappa)^{-2}$, suppose Steps 1 and 3 set

$$s_1 := \min \{ \lceil n^\alpha \rceil, n-1 \}, \tag{4.14}$$

$$\bar{l} := \min \left\{ l : r^{2l} s_1 \geq n \right\} = \left\lceil \ln(n/s_1)/\ln r^2 \right\rceil, \tag{4.15}$$

$$s_{l+1} := \min \left\{ r^{2l} s_1, n \right\} = \min \left\{ r^2 s_l, n \right\}. \tag{4.16}$$

Note that (4.14)–(4.16) yield $s_{l+1} = r^{2l}s_1$ if $l < \bar{l}$, $s_{\bar{l}+1} = n > r^{2(\bar{l}-1)}s_1$. It is easy to see that the proof of Theorem 4.1 covers this modification.

The final iteration $\bar{l}$ doesn't need sampling, since $S_{\bar{l}+1} = X$. Hence, to reduce the sampling costs, we may wish to ensure that $s_{\bar{l}}$, the number of sampled elements, is at most a fixed fraction $\bar{\eta} \in (1/r^2, 1]$ of n when n is large. To this end, suppose that for

$$n \geq \max \left\{ [\, r^2/(\bar{\eta} r^2 - 1) \,]^{1/\alpha}, 3 \right\} \quad \text{with} \quad \bar{\eta} \in (1/r^2, 1], \tag{4.17}$$

we replace (4.14)–(4.15) by

$$\bar{l} := \min \left\{ l : r^{2l} n^\alpha \geq n \right\} = \left\lceil (1-\alpha) \ln n / \ln r^2 \right\rceil, \tag{4.18}$$

$$s_1 := \left\lceil n/r^{2\bar{l}} \right\rceil. \tag{4.19}$$

Then $n^\alpha/r^2 \le s_1 \le \lceil n^\alpha \rceil < n$ replaces (4.14), (4.15) remains true and

$$s_{\bar{l}} < \eta n \quad \text{with} \quad \eta := r^{-2} + n^{-\alpha} \le \bar{\eta}. \tag{4.20}$$

Indeed, $n \le r^{2\bar{l}} n^\alpha < r^2 n$ implies $n^\alpha/r^2 \le s_1 \le \lceil n^\alpha \rceil$; since $n^\alpha \le n^{1/2} \le n-1$ for $n \ge 3$, we have $\lceil n^\alpha \rceil < n$. Next, $n/r^{2\bar{l}} > n^\alpha/r^2 = 1/(\eta r^2 - 1)$ yields $\eta n/r^{2(\bar{l}-1)} > n/r^{2\bar{l}} + 1 > s_1$; thus $\eta n > r^{2(\bar{l}-1)} s_1$. But $n^\alpha \ge r^2/(\bar{\eta} r^2 - 1)$ implies $\eta \le \bar{\eta} \le 1$, so $r^{2(\bar{l}-1)} s_1 < n$, (4.15) holds and (4.16) gives $s_{\bar{l}} < \eta n$. In effect, Theorem 4.1 holds for this modification.

4.3 Using smaller rank gaps

Although the gaps g_l of (2.1) give useful high probability bounds (cf. §5), in practice the average performance on small problems improves for the smaller gaps

$$g_l := (\beta s_l \ln s_l)^{1/2} \quad \text{for } l \le \bar{l}. \tag{4.21}$$

Assuming $\beta > \frac{1}{4}(1-\kappa)^{-2}$, we now sketch briefly how to extend the previous results. First, $\psi(s) := [1 - \kappa(1 + \ln r^2/\ln s)^{1/2}]^2$ replaces $(1-\kappa)^2$ in the relations of §3.2, and (3.7) becomes

$$\mathrm{P_{fail}} := \mathrm{P_{fail}}(s) := 2e^{-2g^2/s} + 2e^{-2\psi(s)g^2/s} = 2s^{-2\beta} + 2s^{-2\beta\psi(s)} \le 4s^{-2\beta\psi(s)}. \tag{4.22}$$

For $\bar{n}$ such that $2\beta\psi(s_1) \ge 1/2$ for all $n \ge \bar{n}$, (4.7)–(4.8) now involve $\mathrm{P_{fail}}(s_l) \le 4 s_l^{-1/2}$, so (4.9) is modified accordingly, whereas (4.11) and (4.13) are replaced by

$$\sum_{l=1}^{\bar{l}} s_{l+1} \mathrm{P_{fail}}(s_l)/4 \le \sum_{l=1}^{\bar{l}-1} r^2 s_l^{1/2} + n^{1/2} r < r^2 \frac{n^{1/2} - s_1^{1/2}}{r-1} + n^{1/2} r < \frac{2r^2 - 1}{r-1} n^{1/2}, \tag{4.23}$$

$$\sum_{l=1}^{\bar{l}} f(s_{l+1}) \mathrm{P_{fail}}(s_l) \le \phi(r^2 s_1) \sum_{l=1}^{\bar{l}} s_{l+1} \mathrm{P_{fail}}(s_l) < 4 \frac{2r^2 - 1}{r-1} \phi(r^2 s_1) n^{1/2}. \tag{4.24}$$

Modify the third terms of (4.4)–(4.5) to complete the proof of Theorem 4.1 as before.

4.4 Handling small subfiles

Since the sampling efficiency decreases when X shrinks, consider the following modification. For a fixed cut-off parameter $n_{\mathrm{cut}} \ge 1$, let sSelect(X, k) be a "small-select" routine that finds the kth smallest element of X in at most $C_{\mathrm{cut}} < \infty$ comparisons when $|X| \le n_{\mathrm{cut}}$ (even bubble sort will do). Then SELECT is modified to start with the following

Step 0 (*Small file case*). If $n := |X| \le n_{\mathrm{cut}}$, return sSelect$(X, k)$.

Our preceding results remain valid for this modification. In fact it suffices if C_{cut} bounds the *expected* number of comparisons of sSelect(X, k) for $n \le n_{\mathrm{cut}}$. For instance, (4.1) holds for $n \le n_{\mathrm{cut}}$ and $\gamma \ge C_{\mathrm{cut}}$, and by induction as in Rem. 2.2(c) we have $C_{nk} < \infty$ for all n, which suffices for the proof of Theorem 4.1.

5 Analysis of nonrecursive versions

Consider a *nonrecursive version* of SELECT in which Steps 2 and 5, instead of SELECT, employ a linear-time routine (e.g., PICK [BFP+72]) that finds the ith smallest of m elements in at most $\gamma_P m$ comparisons for some constant $\gamma_P > 2$.

Theorem 5.1. *Let c_{nk} denote the number of comparisons made by the nonrecursive version of* SELECT, *using* (4.14)–(4.16). *Then for $n \geq 6$, we have*

$$\mathrm{P}\left[\, c_{nk} \leq n + \min\{\, k, n-k \,\} + \hat{\gamma}_P f(n) \,\right] \geq 1 - \bar{l}\mathrm{P}_{\mathrm{fail}} \quad \textit{with} \tag{5.1a}$$

$$\hat{\gamma}_P := 2\gamma_P + \frac{2r^2 - r}{r-1}(3 + 8\gamma_P)\beta^{1/2}, \tag{5.1b}$$

$$\bar{l}\mathrm{P}_{\mathrm{fail}} \leq 4\left\lceil (1-\alpha)\ln n / \ln r^2 \right\rceil n^{-2(1-\kappa)^2\beta}. \tag{5.1c}$$

In particular, $\bar{l}\mathrm{P}_{\mathrm{fail}} = o(n^{-1})$ if $\beta > \frac{1}{2}(1-\kappa)^{-2}$. Moreover,

$$\mathrm{E}c_{nk} \leq n + \min\{\, k, n-k \,\} + \bar{\gamma}_P f(n) \quad \textit{with} \tag{5.2a}$$

$$\bar{\gamma}_P := \hat{\gamma}_P + 4\left(\frac{2r^2 - 1}{r^2 - 1}2\gamma_P + 1/2\right) n^{1/2 - 2(1-\kappa)^2\beta} \ln^{-1/2} n. \tag{5.2b}$$

In particular, $\bar{\gamma}_P \leq \hat{\gamma}_P + 16\gamma_P + 2$ if $\beta \geq \frac{1}{4}(1-\kappa)^{-2}$.

Proof. The cost of Step 2 is at most $2\gamma_P s_1$, with $s_1 \leq \lceil n^{1/2} \rceil \leq f(n) \leq n-1$, since $n \geq 6$. For $\bar{\theta} := \min\{\theta, 1-\theta\}$, the cost of Steps 4 and 5 at iteration l is at most

$$\bar{C}_l := \left(1 + \bar{\theta}\right)(s_{l+1} - s_l) + 3g_l s_{l+1}/s_l + 2\gamma_P \cdot 4g_l s_{l+1}/s_l \tag{5.3}$$

with probability at least $1 - \mathrm{P}_{\mathrm{fail}}$ by (3.6) and Cor. 3.6. Hence c_{nk} exceeds

$$\bar{C} := 2\gamma_P s_1 + \sum_{l=1}^{\bar{l}} \bar{C}_l = 2\gamma_P s_1 + \left(1 + \bar{\theta}\right)(n - s_1) + (3 + 8\gamma_P)\sum_{l=1}^{\bar{l}} g_l s_{l+1}/s_l$$

with probability at most $\bar{l}\mathrm{P}_{\mathrm{fail}}$. But $\bar{C} \leq n + \min\{k, n-k\} + \hat{\gamma}_P f(n)$ by (4.10) and (5.1b), so (5.1a) follows. Then (3.7) and (4.15) with $s_1 \geq n^{\alpha}$ yield (5.1c).

Similarly, $\mathrm{E}c_{nk} \leq 2\gamma_P s_1 + \sum_{l=1}^{\bar{l}}(\mathrm{E}c_l + 2\gamma_P \mathrm{E}\hat{s}_l)$; bounding these costs as for (4.7)–(4.8) via (4.3), (4.10) and (4.11) gives (5.2). □

Remarks 5.2. (a) The bound (5.2) holds if Steps 2 and 5 employ a routine (e.g., FIND [Hoa61]) for which the *expected* number of comparisons to find the ith smallest of m elements is at most $\gamma_P m$ (then $\mathrm{E}c_{nk}$ is bounded as before).

(b) Suppose Step 5 returns to Step 2 if $\hat{s}_l \geq 4g_l s_{l+1}/s_l$. By Cor. 3.6, such loops are finite wp 1, and don't occur with high probability, for n large enough.

(c) Suppose Steps 2 and 5 simply sort S and $\hat{S}_u \cup \hat{S}_v$ by any algorithm that takes at most $\gamma_S m \ln m$ comparisons to sort m elements for a constant γ_S. Then the cost of Step 2 is at most $\gamma_S s_1 \ln n$, because $s_1 < n$; hence $\gamma_S \ln n$ may replace $2\gamma_P$ in (5.1b). Similarly, $\gamma_S \ln n$ replaces γ_P in (5.3) and (5.2b), and $4\gamma_S \ln n$ replaces $8\gamma_P$ in (5.1b). In other words, $n^{1/2}\ln^{3/2} n$ replaces $f(n)$ in (5.1a) and (5.2a) for suitably redefined $\hat{\gamma}_P$ and $\bar{\gamma}_P$.

6 Experimental results

6.1 Implemented algorithms

An implementation of SELECT was programmed in Fortran 77 and run on a notebook PC (Pentium 4M 2 GHz, 768 MB RAM) under MS Windows XP. The input set X was specified as a double precision array, and the partitioning schemes of [Kiw03b, §6] were used. For efficiency, small arrays with $n \le n_{\rm cut}$ were handled by sSelect (cf. §4.4), which typically required less than $3.5n$ comparisons. We used $n_{\rm cut} = 600$ as proposed in [FlR75a], $\alpha = 0.5$, $\beta = 0.3$ in (4.21), $r = 12$ and $\bar{\eta} = 2/r^2$; future work should test other parameters.

6.2 Testing examples

As in [Kiw03b], we used minor modifications of the input sequences of [Val00]:

random A random permutation of the integers 1 through n.

onezero A random permutation of $\lceil n/2 \rceil$ ones and $\lfloor n/2 \rfloor$ zeros.

sorted The integers 1 through n in increasing order.

organpipe The integers $(1, 2, \ldots, n/2, n/2, \ldots, 2, 1)$.

For each input sequence, its (lower) median element was selected for $k := \lceil n/2 \rceil$. To save space, we only add that the results for the twofaced, rotated and m3killer sequences of [Kiw03b] were similar to those of the random, sorted and organpipe inputs, respectively.

6.3 Computational results

We varied the input size n from 50,000 to 16,000,000. For the random and onezero sequences, for each input size, 20 instances were randomly generated; for the deterministic sequences, 20 runs were made to measure the solution time.

The performance of SELECT is summarized in Table 6.1, where the average, maximum and minimum solution times are in milliseconds, and the comparison counts are in multiples of n; e.g., column six gives $C_{\rm avg}/n$, where $C_{\rm avg}$ is the *average number of comparisons* made over all instances. Thus $\gamma_{\rm avg} := (C_{\rm avg} - 1.5n)/f(n)$ estimates the constant γ in the bound (4.1); moreover, for large n we have $C_{\rm avg} \approx 1.5L_{\rm avg}$, where $L_{\rm avg}$ is the average sum of sizes of partitioned arrays. Further, $P_{\rm avg}$ is the average number of SELECT partitions, whereas $N_{\rm avg}$ is the average number of calls to sSelect and $p_{\rm avg}$ is the average number of sSelect partitions per call; both $P_{\rm avg}$ and $N_{\rm avg}$ grow slowly with $\ln n$. Finally, $s_{\rm avg}$ is the *average number of sampled elements*; as predicted by (4.20), $s_{\rm avg}/n$ is about $r^{-2} \approx 0.69\%$ for large n. The average solution times grow linearly with n (except for small inputs whose solution times couldn't be measured accurately), and the differences between maximum and minimum times are quite small (and also partly due to the operating system). Except for the smallest inputs, the maximum and minimum numbers of comparisons are quite close, and $C_{\rm avg}$ nicely approaches the theoretical lower bound of $1.5n$; this is reflected in the values of $\gamma_{\rm avg}$ (which are amazingly stable). The results for the onezero inputs agree completely with our theoretical predictions.

Table 6.1: Performance of SELECT on randomly generated inputs.

Input	Size	Time [msec]			Comparisons [n]			γ_{avg}	L_{avg}	P_{avg}	N_{avg}	p_{avg}	s_{avg}
	n	avg	max	min	avg	max	min		[n]	[$\ln n$]	[$\ln n$]		[%n]
random	50K	2	10	0	1.89	2.05	1.80	26.52	1.23	0.40	0.90	5.50	1.13
	100K	3	10	0	1.79	1.85	1.70	26.61	1.17	0.41	0.91	5.50	0.89
	500K	12	20	10	1.64	1.66	1.60	26.93	1.08	0.58	1.16	5.74	0.81
	1M	24	30	20	1.60	1.61	1.58	26.61	1.06	0.64	1.29	5.83	0.76
	2M	44	50	40	1.57	1.58	1.56	26.96	1.04	0.68	1.41	5.81	0.73
	4M	87	90	80	1.55	1.56	1.54	26.63	1.03	0.69	1.45	6.26	0.72
	8M	167	171	160	1.54	1.54	1.53	25.81	1.02	0.75	1.55	5.98	0.71
	16M	331	341	330	1.53	1.53	1.52	26.75	1.01	0.82	1.70	6.12	0.71
onezero	50K	1	11	0	1.50	1.50	1.50	0.01	1.00	0.18	0.14	1.10	0.86
	100K	4	10	0	1.50	1.50	1.50	0.02	1.03	0.18	0.15	1.14	0.74
	500K	15	20	10	1.50	1.50	1.50	0.00	1.00	0.16	0.15	1.18	0.72
	1M	29	31	20	1.50	1.50	1.50	0.01	1.00	0.14	0.14	1.35	0.71
	2M	58	61	50	1.50	1.50	1.50	0.01	1.00	0.14	0.14	1.30	0.70
	4M	118	121	110	1.50	1.50	1.50	0.01	1.00	0.13	0.13	1.25	0.69
	8M	234	241	230	1.50	1.50	1.50	0.01	1.00	0.13	0.13	1.25	0.69
	16M	470	471	461	1.50	1.50	1.50	0.02	1.00	0.19	0.18	1.15	0.70
sorted	50K	1	10	0	1.89	2.22	1.75	26.45	1.26	0.41	0.91	5.97	1.15
	100K	2	10	0	1.80	1.87	1.64	28.32	1.18	0.42	0.92	6.16	0.90
	500K	8	11	0	1.64	1.66	1.61	26.84	1.08	0.60	1.20	6.00	0.81
	1M	14	20	10	1.60	1.61	1.58	26.41	1.05	0.66	1.32	5.94	0.76
	2M	26	30	20	1.58	1.59	1.57	27.96	1.04	0.68	1.41	5.89	0.73
	4M	47	51	40	1.55	1.56	1.54	26.72	1.03	0.69	1.45	6.17	0.72
	8M	91	100	90	1.54	1.54	1.53	25.89	1.02	0.73	1.53	6.02	0.71
	16M	179	190	170	1.53	1.53	1.52	26.03	1.01	0.83	1.71	6.19	0.71
organpipe	50K	0	0	0	1.90	2.18	1.81	26.85	1.24	0.40	0.89	5.17	1.15
	100K	2	10	0	1.78	1.88	1.71	26.20	1.17	0.41	0.90	5.82	0.89
	500K	8	10	0	1.64	1.67	1.61	27.19	1.08	0.58	1.16	5.85	0.81
	1M	16	20	10	1.60	1.61	1.59	26.05	1.06	0.64	1.29	5.88	0.76
	2M	31	40	30	1.57	1.58	1.55	26.99	1.04	0.67	1.40	6.08	0.73
	4M	59	61	50	1.55	1.56	1.54	25.59	1.03	0.69	1.44	6.05	0.72
	8M	116	121	110	1.54	1.54	1.53	26.63	1.02	0.71	1.49	6.23	0.71
	16M	228	240	220	1.53	1.53	1.52	25.67	1.01	0.83	1.71	5.96	0.71

For our parameters $\alpha = 0.5$ and $\bar{\eta} = 2/r^2$, the test (4.17) is equivalent to $n \geq r^4$, so (4.14) operates only for small $n < r^4 = 20{,}736$. Table 6.2 highlights the danger of choosing s_1 by (4.14) alone (note that for $\bar{\eta} = 1.000001r^{-2}$, (4.17) couldn't hold, being equivalent to $n \geq 10^{12}r^4$). Although s_{avg} increased quite dramatically (cf. Tab. 6.1), C_{avg} decreased slightly for larger n only, γ_{avg} was less stable and the computing times grew significantly; similar deteriorations occured for other inputs.

Although it is not clear how to implement the theoretical scheme of Knuth [Knu98, Ex. 5.3.3–24], we tried to emulate it by using $r^2 = 2$ and (4.21) replaced for $l \leq \bar{l}$ by

$$g_l := \left(\min\{\theta, 1-\theta\} s_l \ln s_l\right)^{1/2}. \tag{6.1}$$

Relative to Tab. 6.1, this scheme made about 3% more comparisons for small n, but was

Table 6.2: Performance of SELECT with $\bar{\eta} = 1.000001/r^2$ on random inputs.

Input	Size	Time [msec]			Comparisons [n]			γ_{avg}	L_{avg}	P_{avg}	N_{avg}	p_{avg}	s_{avg}
	n	avg	max	min	avg	max	min		[n]	[$\ln n$]	[$\ln n$]		[%n]
random	50K	7	11	0	2.03	2.10	1.94	35.84	1.35	0.66	1.41	6.43	64.99
	100K	11	20	10	1.82	1.89	1.76	29.49	1.21	0.65	1.38	6.48	45.92
	500K	41	50	40	1.62	1.64	1.60	22.69	1.07	0.77	1.62	6.37	20.48
	1M	70	91	60	1.58	1.59	1.56	20.64	1.05	0.80	1.66	6.37	14.45
	2M	106	111	100	1.55	1.56	1.54	18.75	1.03	0.87	1.81	6.10	10.22
	4M	175	181	170	1.54	1.54	1.53	19.07	1.02	1.14	2.34	6.27	7.94
	8M	292	301	290	1.53	1.53	1.52	18.87	1.02	1.32	2.70	6.17	5.81
	16M	498	501	491	1.52	1.52	1.52	18.42	1.01	1.34	2.75	6.40	4.03

about 9.5 times slower due to the random sampling overheads (with s_{avg} between 52% and 57%). Eliminating randomization gave the results of Table 6.3. Not suprisingly, this

Table 6.3: Performance of SELECT with Knuth's gap (6.1) and no randomization.

Input	Size	Time [msec]			Comparisons [n]			γ_{avg}	L_{avg}	P_{avg}	N_{avg}	p_{avg}
	n	avg	max	min	avg	max	min		[n]	[$\ln n$]	[$\ln n$]	
random	50K	4	10	0	1.99	2.15	1.87	32.98	1.42	3.35	6.08	5.18
	100K	4	10	0	1.86	2.09	1.77	33.13	1.31	4.40	7.95	4.95
	500K	15	20	10	1.67	2.01	1.63	32.55	1.14	7.09	12.65	5.01
	1M	33	41	30	1.67	2.01	1.59	44.80	1.15	8.84	15.49	5.03
	2M	60	70	50	1.61	1.81	1.56	39.10	1.09	9.23	16.57	5.29
	4M	118	121	110	1.57	1.67	1.55	33.66	1.06	12.51	21.86	5.08
	8M	244	300	240	1.55	1.81	1.53	34.39	1.04	13.95	24.56	5.16
	16M	493	601	460	1.58	1.81	1.52	81.48	1.08	18.07	30.75	5.09
onezero	8M	297	301	290	1.50	1.50	1.50	0.09	1.00	1.45	0.19	1.15
	16M	582	591	580	1.50	1.50	1.50	0.11	1.00	1.45	0.18	1.10
sorted	50K	23	30	20	46.19	46.19	46.19	***	39.86	216.3	366.0	5.18
	100K	56	61	50	56.16	56.16	56.16	***	48.59	471.0	776.0	5.16
	500K	410	421	400	85.83	85.83	85.83	***	75.16	***	***	5.37
	8M	13625	13690	13579	***	***	***	***	147.7	***	***	5.29
	16M	32095	32186	31986	***	***	***	***	175.7	***	***	5.42
organpipe	8M	7238	7281	7200	81.08	81.08	81.08	***	71.59	***	***	5.06
	16M	16486	16564	16453	90.76	90.76	90.76	***	80.55	***	***	5.18

scheme performed fairly well on the random inputs, but quite badly on the deterministic inputs (where "***" denote values exceeding the printout format).

Finally, comparing Tab. 6.1 with [Kiw03b, Tabs. 7.1–7.2], we add that SELECT was slightly slower than its counterpart of [Kiw03b], although the numbers of comparisons made were similar for large n. In fact for small inputs, the ternary version of [Kiw04] made fewest comparisons. The experimental results of [Kiw03a, Kiw03b] suggest that SELECT can compete successfully with refined implementations of quickselect.

Acknowledgment. I would like to thank Olgierd Hryniewicz, Roger Koenker, Ronald

L. Rivest and John D. Valois for useful discussions.

References

[BFP+72] M. R. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest and R. E. Tarjan, *Time bounds for selection*, J. Comput. System Sci. **7** (1972) 448–461.

[Chv79] V. Chvátal, *The tail of the hypergeometric distribution*, Discrete Math. **25** (1979) 285–287.

[CuM89] W. Cunto and J. I. Munro, *Average case selection*, J. of the ACM **36** (1989) 270–279.

[DHUZ01] D. Dor, J. Håstad, S. Ulfberg and U. Zwick, *On lower bounds for selecting the median*, SIAM J. Discrete Math. **14** (2001) 299–311.

[DoZ99] D. Dor and U. Zwick, *Selecting the median*, SIAM J. Comput. **28** (1999) 1722–1758.

[DoZ01] ———, *Median selection requires* $(2+\epsilon)N$ *comparisons*, SIAM J. Discrete Math. **14** (2001) 312–325.

[FlR75a] R. W. Floyd and R. L. Rivest, *The algorithm SELECT—for finding the* i*th smallest of* n *elements (Algorithm 489)*, Comm. ACM **18** (1975) 173.

[FlR75b] ———, *Expected time bounds for selection*, Comm. ACM **18** (1975) 165–172.

[GeS96] A. V. Gerbessiotis and C. J. Siniolakis, *Concurrent heaps on the BSP model*, Tech. Report PRG-TR-14-96, Oxford University Computing Lab., Oxford, UK, 1996.

[Grü99] R. Grübel, *On the median-of-k version of Hoare's selection algorithm*, Theor. Inform. Appl. **33** (1999) 177–192.

[Hoa61] C. A. R. Hoare, *Algorithm 65:* FIND, Comm. ACM **4** (1961) 321–322.

[Hoe63] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc. **58** (1963) 13–30.

[Kiw02] K. C. Kiwiel, *Randomized selection revisited*, Tech. report, Systems Research Institute, Warsaw, 2002. Available at the URL http://arxiv.org/abs/cs.DS/0204033.

[Kiw03a] ———, *Partitioning schemes for quicksort and quickselect*, Tech. report, Systems Research Institute, Warsaw, 2003. Available at the URL http://arxiv.org/abs/cs.DS/0312054.

[Kiw03b] ———, *Randomized selection with quintary partitions*, Tech. report, Systems Research Institute, Warsaw, 2003. Available at the URL http://arxiv.org/abs/cs.DS/0312055.

[Kiw04] ———, *Randomized selection with tripartitiong*, Tech. report, Systems Research Institute, Warsaw, 2004. Available at the URL http://arxiv.org/abs/cs.DS/0401003.

[KMP97] P. Kirschenhofer, C. Martínez and H. Prodinger, *Analysis of Hoare's* FIND *algorithm with median-of-three partition*, Random Stuctures and Algorithms **10** (1997) 143–156.

[Knu97] D. E. Knuth, *The Art of Computer Programming. Volume I: Fundamental Algorithms*, third ed., Addison-Wesley, Reading, MA, 1997.

[Knu98] ———, *The Art of Computer Programming. Volume III: Sorting and Searching*, second ed., Addison-Wesley, Reading, MA, 1998.

[Kor78] V. S. Koroliuk, ed., *Handbook on Probability Theory and Mathematical Statistics*, Naukova Dumka, Kiev, 1978 (Russian).

[PRKT83] J. T. Postmus, A. H. G. Rinnooy Kan and G. T. Timmer, *An efficient dynamic selection method*, Comm. ACM **26** (1983) 878–881.

[SPP76] A. Schönhage, M. Paterson and N. Pippenger, *Finding the median*, J. Comput. System Sci. **13** (1976) 184–199.

[Val00] J. D. Valois, *Introspective sorting and selection revisited*, Software–Practice and Experience **30** (2000) 617–638.