A probabilistic analysis of the Floyd-Rivest expected time selection algorithm

Alexandros V. Gerbessiotis
CS Department
New Jersey Institute of Technology
Newark, NJ 07102,
USA.

Constantinos J. Siniolakis
The American College of Greece
6 Gravias St.
Aghia Paraskevi
Athens, 15342
Greece.

November 27, 2002

Abstract

In this work we present a probabilistic analysis of the Floyd-Rivest expected time selection algorithm. In particular we show that a refinement of the bootstrapped version of the Floyd-Rivest algorithm that determines the C-th order statistic by performing an expected $n + C + O(n^{1/2})$ number of comparisons can be made into a randomized algorithm that performs $n + C + O(n^{1/2} \log^{3/2} n)$ comparisons with probability at least $1 - 1/n^{\rho}$, for any constant $\rho > 0$.

Keywords. Randomization - Algorithms - Selection - Floyd-Rivest selection algorithm

1 Introduction

The selection problem that is, the computation of the C-th smallest key from a set of n keys has always been an interesting research topic [2, 12, 13]. Median finding in linear time is examined in [2, 17] and lower bounds for various values of C are also presented in [2, 11, 12, 15]. The minimum average number of comparisons required to find the C-th largest of n keys has also drawn considerable interest. A lower bound of $n + C - O(1)$ comparisons, $1 \leq C \leq \lceil n/2 \rceil$, is presented in [4], which matches the performance, within low order terms, of the Floyd-Rivest algorithm [7]. In particular in [7] two algorithms are presented one whose expected number of comparisons is $n + C + O(n^{2/3} \log^{1/3} n)$ and a bootstrapped version of this algorithm with expected number of comparisons equal to $n + C + O(n^{1/2})$. In the remainder $\log n$ denotes the natural logarithm of n and $\lg n$ the base two logarithm of n. In [9] a sequential randomized algorithm for median finding that can be further extended to a parallel one is proposed. The sequential version of the randomized algorithm requires $3n/2 + O(n^{2/3} \log^{1/3} n)$ comparisons, with high probability. In this work we generalize the oversampling techniques used in [9] and present a randomized selection algorithm that fine-tunes the bootstrapped version of the Floyd-Rivest selection algorithm. Our augmentation uses random oversampling [16] and determines the C-th smallest of a set of n keys in $n + C + o(n)$ comparisons with probability at least $1 - 1/n^{\rho}$, for any constant $\rho > 0$. The low order term describing the number of comparisons performed is $O(n^{1/2} \log^{3/2} n)$, comparable to the $O(n^{1/2})$ term of the expected time Floyd-Rivest algorithm. Without loss of generality C is such that $C \leq n/2$. By symmetry, the algorithm also establishes the C-th largest of n keys, which is the $(n - C)$-th smallest.

2 Random Sampling

In this section we briefly present or cite some tight results on random oversampling [16]. The use of oversampling parameter s to choose a sample of size $ks - 1$ uniformly at random without replacement out of a set of N keys was used in [8] in the case of parallel sorting. In particular it was shown in [8] that the $k - 1$ equidistant splitters of a sorted sample of size $ks - 1$ split N keys into k "buckets" of approximately the same size $(1 + o(1))N/k$. The result is summarized below.

Lemma 1 *Let $k \geq 2$, $s \geq 1$, $ks < N/2$, $n \geq 1$, $0 < \epsilon < 1$, $\rho > 0$, and*

$$s \geq \frac{1+\epsilon}{\epsilon^2} \left(2\rho \log n + \log \left(2\pi k^2 (ks-1) e^{1/(3(ks-1))} \right) \right).$$

Then the probability that any one of the k sets induced by the $k - 1$ splitters is of size more than $\lceil (1+\epsilon)\frac{N-k+1}{k} \rceil$ is at most $1/n^{\rho}$.

For selection algorithms we need only split X into three sets. We introduce some relevant notation in this paragraph. Let $X = \langle x_1, \ldots, x_N \rangle$ be a set of input keys ordered such that $x_i < x_{i+1}$, for $1 \leq i \leq N - 1$ assuming that all keys are distinct. We are interested in finding the C-th smallest of the keys in set X and without loss of generality $C \leq \lceil N/2 \rceil$ since otherwise, we can use symmetric arguments to find the $(N-C)$-th largest key. Let then be $c = C/N$. For integers $\lambda \geq 1$ and $s \geq 1$, let $Y = \langle y_1, \ldots, y_{2\lambda s-1} \rangle$ be a randomly chosen subset of $2\lambda s - 1 \leq N/2$ keys of X, also ordered so that $y_i < y_{i+1}$. The following three subsets are then formed for some positive integers μ_0, μ_2, and $t \geq 1$ such that $2c\lambda s > \mu_0 t/2$.

$$
\begin{aligned}
X_0 &= \{x \mid x \in X \text{ and } x < y_{2c\lambda s - \mu_0 t/2}\}, \\
X_1 &= \{x \mid x \in X \text{ and } y_{2c\lambda s - \mu_0 t/2} < x < y_{2c\lambda s + \mu_2 t/2}\}, \\
X_2 &= \{x \mid x \in X \text{ and } y_{2c\lambda s + \mu_2 t/2} < x\}.
\end{aligned}
$$

For $0 \leq i \leq 2$, let $n_i = |X_i|$ and let $p_i(j)$ be the probability that $n_i = j$. The following has been shown in [8].

Lemma 2 *For $i = 0, 2$, probabilities $p_i(j)$ are given by the following expressions*

$$
p_0(j) = \frac{\binom{j}{2c\lambda s - \mu_0 t/2 - 1}\binom{N-j-1}{2(1-c)\lambda s + \mu_0 t/2 - 1}}{\binom{N}{2\lambda s - 1}},
$$

and

$$
p_2(j) = \frac{\binom{j}{2(1-c)\lambda s - \mu_2 t/2 - 1}\binom{N-j-1}{2c\lambda s + \mu_2 t/2 - 1}}{\binom{N}{2\lambda s - 1}},
$$

The bootstrapped Floyd-Rivest algorithm splits an input key set into three sets like X_0, X_1, X_2. By establishing lower and upper bounds on the sizes of X_0 and X_2 one can guarantee that the required statistic is located in the middle set X_1 with high probability; in the remainder the term "with high probability" would mean with probability at least $1 - 1/n^\rho$ for some constant $\rho > 0$. The bounds for the size of one of X_0, X_2 lead by symmetry to corresponding bounds for the other set. In particular we first show that with high probability X_0 is of size at least $cN - f(N)$. We then show that X_0 is with high probability at most $cN - 1 = C - 1$. By symmetry we can also show that with high probability X_2 is of size at least $(1-c)N - f(N)$ and at most $(1-c)N - 1$. The always positive function $f(N)$ such that $f(N) < N$ is fixed within the algorithm.

Let $B = cN - f(N) - 1$. First we would like to show that the term $\sum_{j \leq B} p_i(j)$, $i = 0$, is sufficiently small for appropriate choices of $f(N)$ thus establishing a lower bound on the size of X_0. The following was proved for the case of median-finding in [9] for the special case $c = 1/2$.

Lemma 3 *Let $\lambda, \mu \geq 1$, $s, t \geq 1$, $2\lambda s - 1 < N/2$, $n \geq N$, $\lambda s > \mu t/2$, and $\rho > 0$. If $f(N) = (N\ \mu t)/(2\ \lambda s)$ and $\mu t = 2\sqrt{3\rho\ \lambda s\ \log n}$, the probability that any one of the X_0, X_2 is of size less than $N/2 - f(N)$ is at most $\sqrt{2\pi(2\lambda s - 1)}\ e^{1/(6(2\lambda s-1))}\ /n^\rho$.*

A direct generalization of Lemma 3 and its proof leads to the following, that provides a lower bound for X_0.

Proposition 1 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda, \mu \geq 1$, $s, t \geq 1$, $2\lambda s - 1 < N/2$, $n \geq N$, $2c\lambda s > \mu t/2$, $\mu t \geq 6$, and $\rho > 1$. If $f(N) = (N\ \mu t)/(2\ \lambda s)$ and $\mu t = 2\sqrt{3\rho\ 2c\ \lambda s\ \log n}$, then the probability that set X_0 is of size more than $cN - f(N) - 1$ is at least $1 - n^{1-\rho}$.*

Proof The proof of Proposition 1 is similar to that of Lemma 3. An outline is shown for completeness in the appendix. ∎

A lower bound for X_2 can be found by symmetry by noting that in $p_0(j)$ and $p_2(j)$ the c in the former becomes $1-c$ in the latter and μ_0 becomes μ_2.

Proposition 2 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda, \mu \geq 1$, $s, t \geq 1$, $2\lambda s - 1 < N/2$, $n \geq N$, $2(1-c)\lambda s > \mu t/2$, $\mu t \geq 6$, and $\rho > 1$. If $f(N) = (N\ \mu t)/(2\ \lambda s)$ and $\mu t = 2\ \sqrt{3\rho\, 2(1-c)\ \lambda s\ \log n}$, then the probability that set X_2 is of size more than $(1-c)N - f(N) - 1$ is at least $1 - n^{1-\rho}$.*

Proof The proof is identical to the proof of Proposition 1. For the probability bound derivation, instead of requiring the condition $\mu t \geq 6 \geq 4c + 2c \geq 4c + \mu t/(2\lambda s)$, we require that the following condition is true $\mu t \geq 6 \geq 4(1-c) + 2(1-c) \geq 4(1-c) + \mu t/(2\lambda s)$, which, however, leads to the same lower bound for μt of $\mu t \geq 6$. ∎

We can similarly establish upper bounds for the sizes of the sets X_0 and X_2 extending the proof of Lemma 1 proven in [8].

Proposition 3 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda, \mu \geq 1$, $s, t \geq 1$, $2\lambda s - 1 < N/2$, $n \geq N$, $2c\lambda s > \mu t/2$, and $\rho > 1$. If $\mu t \geq 2\ \sqrt{2\rho\, 2c\ \lambda s\ \log n}$ then the probability that set X_0 is of size at most $cN - 1$ is at least $1 - n^{1-\rho}$.*

Proof The proof of the Proposition is a generalization of the proof of Lemma 1 proven in [8]. We do not repeat this proof here; we show however how the probability bound. For $B = cN$ the probability bound we obtain, following the steps of the proof in [8], is

$$\sum_{j \geq B} p_0(j) \leq \sqrt{2\pi(2\lambda s - 1)}\ exp(\frac{1}{6(2\lambda s - 1)})\ exp(-\frac{(Bq - 2c\lambda s + \mu t/2 + 1)^2}{2Bq}).$$

The numerator $(Bq - 2c\lambda s + \mu t/2 + 1)$ has $\mu t/2$ as a lower bound, the denominator $2Bq$ has $4\lambda sc$ as an upper bound. The lower bound for μt leads to the desired result, noting that the expression preceding the exponential term is $O(\sqrt{n})$ and at most n. ∎

An upper bound on X_2 follows by symmetry.

Proposition 4 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda, \mu \geq 1$, $s, t \geq 1$, $2\lambda s - 1 < N/2$, $n \geq N$, $2(1-c)\lambda s > \mu t/2$, and $\rho > 1$. If $\mu t \geq 2\ \sqrt{2\rho\, 2(1-c)\ \lambda s\ \log n}$ then the probability that set X_2 is of size at most $(1-c)N - 1$ is at least $1 - n^{1-\rho}$.*

The following two results and their corollaries will be used in the analysis of the Floyd-Rivest algorithm.

Proposition 5 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda, \mu \geq 1$, $s, t \geq 1$, $2\lambda s - 1 < N/2$, $n \geq N$, $2c\lambda s > \mu t/2$. If $\mu t = 2\ \sqrt{3\rho\, 2c\ \lambda s\ \log n}$, then for any always positive $f(N) < N$ the probability that set X_0 is of size more than $c(N - f(N)) - 1$ is at most $\sqrt{2\pi(2\lambda s - 1)}\ e^{1/(6(2\lambda s - 1))}$ $exp(-((\mu t/2)^2 - (\mu t)(cf(N)/N)(2\lambda s - 1))/(2(2\lambda s - 1)c))$.*

Proof The Proposition satisfies the conditions of Proposition 3 since $\mu t = 2\sqrt{3\rho\, 2c\, \lambda s\ \log n} \geq 2\sqrt{2\rho\, 2c\, \lambda s\ \log n}$. Using a $B = c(N - f(N))$, the lower bound in Proposition 3 for $Bq - 2c\lambda s + \mu t/2 + 1$ becomes $Bq - 2c\lambda s + \mu t/2 + 1 \geq \mu t/2 - c(f(N)/N)(2\lambda s - 1) + 1 - c \geq \mu t/2 - c(f(N)/N)(2\lambda s - 1)$, and the upper bound for Bq is given by $Bq = c(N - f(N))(2\lambda s - 1)/N \leq c(2\lambda s - 1)$, and the result follows noting also that $(a-b)^2 \geq a^2 - 2ab$ for real a, b. ■

By symmetry the following applies to X_2.

Proposition 6 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda, \mu \geq 1$, $s, t \geq 1$, $2\lambda s - 1 < N/2$, $n \geq N$, $2(1-c)\lambda s > \mu t/2$. If $\mu t = 2\sqrt{3\rho\, 2c\, \lambda s\ \log n}$, then for any always positive $f(N) < N$ the probability that set X_2 is of size more than $(1-c)(N - f(N)) - 1$ is at most $\sqrt{2\pi(2\lambda s - 1)}\ e^{1/(6(2\lambda s - 1))}$ $exp(-((\mu t/2)^2 - (\mu t)((1-c)f(N)/N)(2\lambda s - 1))/(2(2\lambda s - 1)(1-c)))$.*

The Propositions can be simplified in the form of the following Corollaries.

Corollary 1 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda \geq 1$, $2\lambda - 1 < N/2$, $n \geq N$, and $\rho > 1$. Let $2c\lambda > \mu_0/2$ and $\mu_0 = 2\sqrt{3\rho\, 2c\, \lambda\ \log n} \geq 3$. Then, with probability at least $1 - n^{1-\rho}$, set X_0 is of size at least $cN - N\sqrt{3\rho\, 2c\ \log n}/\sqrt{\lambda}$.*

Proof The Corollary is a direct consequence of Proposition 1 by substituting for $s = t = 1$ and $\mu = \mu_0 = 2\sqrt{3\rho\, 2c\, \lambda\ \log n}$ in $f(N) = N\mu t/2\lambda s = N\mu/2\lambda$. ■

Corollary 2 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $\lambda \geq 1$, $2\lambda - 1 < N/2$, $n \geq N$, and $\rho > 1$. Let $2(1-c)\lambda > \mu_2/2$ and $\mu_2 = 2\sqrt{3\rho\, 2(1-c)\, \lambda\ \log n} \geq 3$. Then, with probability at least $1 - n^{1-\rho}$, set X_2 is of size at least $N - cN - N\sqrt{3\rho\, 2(1-c)\ \log n}/\sqrt{\lambda}$.*

Proof The claim is a direct consequence of Proposition 2 by substituting for $s = t = 1$ and $\mu = \mu_2 = 2\sqrt{3\rho\, 2(1-c)\, \lambda\ \log n}$ in $f(N) = N\mu t/2\lambda s = N\mu/2\lambda$. ■

Corollary 3 *Let $C \leq \lceil N/2 \rceil$ and $c = C/N$. Let $y_i \geq 36$, $y_i < y_{i+1}/2$, $n \geq N = y_{i+1}$, $y_{i+1} = dy_i$, $d > 38$, and $\rho > 1$. Let $\mu_0 = 2\sqrt{3\rho\, c\, (y_i + 1)\ \log n} \geq 1$ and $c\ (y_i + 1) > \mu_0/2$. Then, with probability at least $1 - n^{1-\rho}$, set X_0 is of size at most $c(y_{i+1} + 1) - \sqrt{3\rho c(y_{i+1} + 1) \log n} - 1$.*

Proof The proof follows directly from proposition 5 with $s = t = 1$, $2\lambda - 1 = y_i$, $N = y_{i+1}$, and $f(n) = \sqrt{3\rho(y_{i+1} + 1) \log n/c}$.

The numerator $((\mu t/2)^2 - (\mu t)(cf(N)/N)(2\lambda s - 1))$ of the probability bound in Proposition 5 is such that $(\mu t/2)^2 = 3\rho(y_i + 1)c \log n$ and $(\mu t)(cf(N)/N)(2\lambda s - 1)) = 6cy_i\rho \log n\sqrt{(y_{i+1} + 1)(y_i + 1)/y_{i+1}^2}$. The restrictions $d > 38$ and $y_i > 36$ are required for inequality $\sqrt{(y_{i+1} + 1)(y_i + 1)/y_{i+1}^2} < 1/6$ to hold, since then $(y_{i+1} + 1)(y_i + 1)/y_i^2 = (dy_i + 1)(y_i + 1)/d^2 y_i^2 = d(y_i + y_i/36)(y_i + y_i/36)/d^2 y_i^2 \leq (37/36)^2/d$ and the latter is less than 36 provided that $d > 38$. which yields a lower bound for $-(\mu t)(cf(N)/N)(2\lambda s - 1)) \geq -cy_i\rho \log n$. Thus the numerator of the error bound is at least $2\rho(y_i + 1)c \log n$. The denominator is $2(2\lambda s - 1)c = 2(y_i + 1)c$. Therefore the exponential probability term becomes at most $exp(-\rho \log n) = 1/n^{\rho}$. The multiplicative term $\sqrt{2\pi(2\lambda s - 1)}\ e^{1/(6(2\lambda s-1))}$ is $O(\sqrt{y_i})$ and at most n. The Corollary then follows. ■

A claim symmetric to claim 3 can also be established for X_2.

3 Randomized Selection

The bootstrapped version of the Floyd-Rivest algorithm works as follows. An initial random sample of the input is chosen and two pivot keys within this sample are also selected so that the C-th smallest key is between them. In the following step a new sample is selected that is a superstep of the sample selected in the previous iteration and two new pivot keys are also selected relative to this sample; the intent of the bootstapped method is to maintain the C-th order statistic within a set controlled by the two pivot keys whose size is also controllable and small compared to n. The algorithm provided in [7] with the given choices for the initial sample size, the expansion factor of the sample set and the number of iterations of the bootstrapped method has an expected comparison count of $n + C + O(n^{1/2})$. By using the results of the previous propositions and corollaries, properly choosing the original sample size and the pivot keys, the expansion factor d and the number of iterations, we derive a randomized selection variant of the Floyd-Rivest algorithm that performs $n+C+O(n^{1/2}\log^{3/2} n)$ comparisons with probability at least $1-1/n^{\rho}$ for any $\rho > 0$.

Before we present algorithm SELECT and describe its performance we present some preliminary results/observations that will be used in the bootstrapping phase. The following key comparison schedule would be used. Let $m < M$ be two keys that split a sample set X into three disjoint sets X_0, X_1, X_2. We can determine the set of keys of X_1 with the following comparison schedule referred to as KEYSCHEDULE.

a. Compare key x to M first. If $x > M$ stop (x is in X_2).

b. Otherwise compare x to m. If $x < m$ stop (x is in X_0).

c. Otherwise x is in X_1.

Let x_0, x_1, x_2 be the sizes of X_0, X_1, X_2 respectively. Then, $x_0 + x_1 + x_2 = x$. Let l_0 and l_2 be lower bounds on the sizes of x_0, x_2 respectively. The size x_1 of X_1 is bounded above by

$$x_1 = x - x_0 - x_2 \quad \leq \quad x - l_0 - l_2. \tag{1}$$

The number of comparisons performed to determine X_1 is equal to $x_2 + 2x_0 + 2x_1$ as a results of the sequence of steps (a), (b), (c) in KEYSCHEDULE. It is immediate that

$$x_2 + 2x_0 + 2x_1 = 2x - x_2 \quad \leq \quad 2x - l_2. \tag{2}$$

Let sample X be augmented into a sample set Y such that $X \subset Y$. Let the size of Y be N. For the keys in $Y - X$ let N_0, N_1, N_2 be the number of keys in $Y - X$ that are less than m_1, between m_1 and M_1, and greater than M_1. We have that $N_0 + N_1 + N_2 = N - x$. Suppose that we know a lower bound for the number of keys in Y greater than M_1 and let this lower bound be L_2. Then we have $x_2 + N_2 \geq L_2$. Let Z be the set of keys whose values are between m_1 and M_1. We estimate the number of comparisons performed to find Z using KeySchedule. Since $X \subset Y$ we already know the position of each key in X relative to m_1 and M_1. Therefore we need to determine the position of elements in $Y - X$ only. Using KeySchedule and the earlier result for X we conclude that the number of comparisons performed to determine the number of keys of Z is equal to

$$N_2 + 2N_0 + 2N_1 = 2(N_0 + N_1 + N_2) - N_2 = 2(N - x) - N_2 \quad \leq \quad 2(N - x) - L_2 + x_2. \qquad (3)$$

Theorem 1 *For any constant $\rho > 1$, Algorithm* Select*, with probability at least $1 - O(n^{1-\rho})$, determines the C-th smallest, $C \leq \lceil n/2 \rceil$, of n keys within $n + C + O(n^{1/2} \log^{3/2} n)$ comparisons with probability at least $1 - 1/n^{\rho}$ for any constant $\rho > 0$.*

Proof A high-level description of Select is given below. In the discussion to follow, we assume that we can sort r keys in $r \log r$ comparisons. If $dr \log r$ comparisons are required for sorting, for some constant d, this will only affect by a multiplicative constant the lower order terms in the analysis to follow. Let X be the input key set of cardinality n. Procedure Select works as follows.
Base case (Iteration 0). At iteration 0, a set $Y_1 \subset X$ of size $|Y_1| = y_1 = m = n^a$, for some constant $\alpha < 1$, is randomly chosen among the keys of the input set. Let m_1 and M_1 be the $(c(y_1+1) - \mu_0^{(1)}/2)$-th and $(c(y_1+1) + \mu_2^{(1)}/2)$-th smallest keys of Y_1 respectively, where $\mu_0^{(1)} = 2\sqrt{3\rho c(y_1+1)\log n}$ and $\mu_2^{(1)} = 2\sqrt{3\rho(1-c)(y_1+1)\log n}$. For $c > 3\rho \log n/(y_1+1)$, m_1 is completely defined; otherwise, we set $m_1 = -\infty$. Pivot keys m_1 and M_1 are selected by sorting set Y_1 in $m \log m$ comparisons (alternatively, we could use a deterministic selection algorithm). The comparison count at iteration 0 is thus $m \log m$.

This completes the base case of the algorithm. The size of the base set Y_1 will be fixed at the completion of the description and the analysis of Select. We then show how the base set is expanded i.e. how the Floyd-Rivest bootstrapping process evolves for the randomized algorithm we present in this work.
Iteration 1. At iteration 1, the sample Y_1 is expanded into a sample Y_2 of size $|Y_2| = y_2 = dm = dy_1$, where $d > 38$ and will be determined at the conclusion of the algorithm, and Y_2 is uniformly at random chosen so that $Y_1 \subset Y_2$. The random selection of $(d-1)m$ additional keys is thus required to expand Y_1 of size m into Y_2 of size dm. We can consider set Y_1 a sample of the larger set Y_2. Let Z_1 be the set of keys of Y_2 between pivot keys m_1 and M_1 and let z_1 be the number of such keys. For the keys in Y_1 we know already its number of keys between m_1 and M_1, less than m_1 and greater than M_1 since we sorted Y_1 to collect m_1 and M_1. So determining Z_1 requires that we consider only the $Y_2 - Y_1$ keys.

We first determine an upper bound for the size z_1 of Z_1 by determining a lower bound for the set of keys less than m_1 and greater than M_1 and then using Eq. 1. By Corrolaries 1 and 2, with sample size equal to $y_1 = m$, i.e. $\lambda = (y_1+1)/2$, key size $N = y_2 = dm$, and using in Corrolary 1 $\mu_0 = \mu_0^{(1)} = 2\sqrt{3\rho c(y_1+1)\log n}$, and Corollary 2 $\mu_2 = \mu_2^{(1)} = 2\sqrt{3\rho(1-c)(y_1+1)\log n}$ we obtain that

$$z_1 \leq (y_2/\sqrt{(y_1+1)/2})(\sqrt{3\rho 2c \log n} + \sqrt{3\rho 2(1-c)\log n}), \tag{4}$$

with probability $1 - O(n^{1-\rho})$.

We then count the number of comparisons we performed to determine set Z_1 and also, z_1. We use KeySchedule to compare elements of $Y_2 - Y_1$ to m_1, M_1 and Eq. 3 to determine an upper bound on the number of comparisons performed. Expression $2(N-x)$ in Eq. 3 is $2(y_2 - y_1) = 2(d-1)m$. Parameter x_2 in Eq. 3 is the number of keys in Y_1 greater than M_1. Since M_1 is the $c(y_1+1)+\mu_2^{(1)}/2$-th smallest key of Y_1 we conclude that x_2 is $y_1 - c(y_1+1) - \mu_2^{(1)}/2$. Since L_2 is a lower bound on the size of the set of keys of Y_2 greater than M_1, the application of Corollary 2 gives that $L_2 = (1-c)y_2 - (y_2/(\sqrt{(y_1+1)/2}))\sqrt{3\rho 2(1-c)\log n}$. Therefore the number of comparisons P_1 performed to determine Z_1 and also, its size z_1 is at most

$$\begin{aligned} P_1 &= 2(d-1)m - (1-c)y_2 + (y_2/(\sqrt{(y_1+1)/2}))\sqrt{3\rho 2(1-c)\log n} + y_1 - c(y_1+1) - \mu_2^{(1)}/2 \\ &= 2(d-1)m - (1-c)dm + (y_2/(\sqrt{(y_1+1)/2}))\sqrt{3\rho 2(1-c)\log n} + m - c(m+1) - \mu_2^{(1)}/2 \\ &\leq (d-1)m + c(d-1)m + (y_2/(\sqrt{(y_1+1)/2}))\sqrt{3\rho 2(1-c)\log n} \end{aligned} \tag{5}$$

After these comparisons are informed we also establish the number of keys in Y_2 less than m_1.

The following steps are preparatory for the next iteration 2 the same way the base case prepared iteration 1. Both m_1 and M_1 were chosen to be the $(c(y_1+1) - \mu_0^{(1)}/2)$-th and $(c(y_1+1) + \mu_2^{(1)}/2)$-th smallest keys of Y_1 considered to be a sample of Y_2. By the result of Corollary 3, for $\mu_0 = \mu_0^{(1)}$ there, we conclude that the number of keys in Y_2 less than m_1 is at most $c(y_2+1) - \sqrt{3\rho c(y_2+1)\log n} - 1$ with high probability. Therefore the $(c(y_2+1) - \mu_0^{(2)}/2)$-th smallest key of Y_2, where $\mu_0^{(2)} = 2\sqrt{3\rho c(y_2+1)\log n}$, is guaranteed, with high probability, to belong to Z_1. Similarly and symmetrically to Corollary 3 we can show that the $(c(y_2+1) + \mu_2^{(2)}/2)$-th smallest key of Y_2 also belongs to Z_1 with high probability, where $\mu_2^{(2)} = 2\sqrt{3\rho(1-c)(y_2+1)\log n}$.

Let m_2 and M_2 be the $(c(y_2+1) - \mu_0^{(2)}/2)$-th and $(c(y_2+1) + \mu_2^{(2)}/2)$-th smallest keys of Y_2. We have just proved that both m_2 and M_2 belong to Z_1 with high probability. If $c > 3\rho \log n/(y_2+1)$, then m_2 is defined, otherwise, we use $m_2 = -\infty$. In order to find m_2, M_2 we sort Z_1 by performing $z_1 \log z_1 \leq z_1 \log n$ comparisons to obtain m_2 and M_2. Given the bound for z_1 the number of Q_1 of comparison performed to sort Z_1 is bounded above by

$$\begin{aligned} Q_1 &= z_1 \lg z_1 = z_1 \lg n \\ &\leq \log n (y_2/\sqrt{(y_1+1)/2}) O(\sqrt{\rho c \log n} + \sqrt{\rho(1-c)\log n}) \\ &\leq (y_2/\sqrt{(y_1+1)/2}) O(\sqrt{\rho c \log^3 n} + \sqrt{\rho(1-c)\log^3 n}). \end{aligned} \tag{6}$$

The total number of comparison performed at iteration 1 is the sum of P_1 and Q_1, and let this total be R_1. We have that R_1 is given by the following quantity.

$$\begin{aligned} R_1 &= P_1 + Q_1 \\ &\leq (d-1)m + c(d-1)m + (y_2/\sqrt{(y_1+1)/2}) O(\sqrt{\rho c \log^3 n} + \sqrt{\rho(1-c)\log^3 n}). \end{aligned} \tag{7}$$

Iteration i: By induction, in the i-th iteration, a set Y_{i+1} of size $|Y_{i+1}| = y_{i+1} = d^i m$ is randomly chosen so that $Y_i \subset Y_{i+1}$, by randomly selecting $d^{i-1}(d-1)m$ additional keys of the input key set. Set Y_i can be considered as a sample for the larger set Y_{i+1}. The keys of Y_{i+1} with values between m_i and M_i form set Z_i and by Corollaries 1 and 2, for $\lambda = (y_i+1)/2$, $\mu_0 = \mu_0^{(i)} = 2\sqrt{3\rho c(y_i+1)\log n}$, $\mu_2 = \mu_2^{(i)} = 2\sqrt{3\rho(1-c)(y_i+1)\log n}$, $y_i = d^{i-1}m$, and $N = y_{i+1} = d^i m$ there, the cardinality z_i of set Z_i is at most

$$z_i \le (y_{i+1}/\sqrt{(y_i+1)/2})(\sqrt{3\rho 2c\log n} + \sqrt{3\rho 2(1-c)\log n}), \tag{8}$$

with probability $1 - O(n^{1-\rho})$.

Set Z_i is found in at most

$$P_i \;\le\; d^{i-1}(d-1)m + cd^{i-1}(d-1)m + (y_{i+1}/(\sqrt{(y_i+1)/2}))\sqrt{3\rho 2(1-c)\log n} \tag{9}$$

comparisons, using KeySchedule. Let m_{i+1} and M_{i+1} be the $(c(y_{i+1}+1)-\mu_0^{i+1}/2)$-th and $(c(y_{i+1}+1)+\mu_2^{i+1}/2)$-th smallest keys of Y_{i+1} that with high probability (and the use of Corollary 3) are guaranteed to be in Z_i, where $\mu_0^{i+1} = 2\sqrt{3\rho c(y_{i+1}+1)\log n}$ and $\mu_2^{i+1} = 2\sqrt{3\rho(1-c)(y_{i+1}+1)\log n}$. Set Z_i is sorted in $Q_i = z_i \log z_i \le z_i \log n$ comparisons and pivot keys m_{i+1} and M_{i+1} can be thus determined. The total number of comparisons performed at iteration i is thus,

$$R_i \;\le\; d^{i-1}(d-1)m + cd^{i-1}(d-1)m + (y_{i+1}/\sqrt{(y_i+1)/2})O(\sqrt{\rho c\log^3 n} + \sqrt{\rho(1-c)\log^3 n}) \tag{10}$$

The bootstrapping process terminates for $i = t$ when $y_{t+1} = d^t m = n$ and thus $Y_{i+1} = X$ and Z_t then contains the C-th smallest key, with high probability. The desired statistic is computed by selecting the $(C-l)$-th smallest key of Z_t, where l is the number of keys of Y_{t+1} less than m_t. Set Z_t is sorted for this task to be performed. The number of comparisons performed at iteration t is thus R_t.

The total cost in terms of number of comparisons performed if we add the $m\log m$ contribution of the base step is is given by

$$\begin{aligned} R &= \sum_{i=0}^{t} R_i = m\log m + \sum_{i=1}^{t} R_i \\ &\le m\log m + \sum_{i=1}^{t} d^{i-1}(d-1)m + cd^{i-1}(d-1)m + O(\frac{d^i m}{\sqrt{(d^{i-1}m+1)/2}}\sqrt{\rho c\log^3 n}) \\ &= m\log m + (d^t-1)m + c(d^t-1)m + \sum_{i}^{t} O(d^{(i+1)/2}\sqrt{m\rho c\log^3 n}) \\ &= n + cn + O(d^{(t+1)/2}\sqrt{m\rho\log^3 n}). \end{aligned}$$

Given that $md^t = n$, $d > 36$, for $d = O(1)$ the number of iterative steps t becomes $t = O(\log n)$.

One way to fix the parameters of the problem i.e. $y_1 = m$, d and the number of iterations t is as follows. Let us fix $m = n^a$, for any constant $a \le 1/2$. Since we require that $md^t = n$, then we deduce that d must be $d = (n/m)^{1/t}$, and the term within the big-Oh expression for R thus becomes, $d^{1/2}n^{1/2}\lg^{3/2} n$ since $d^{t/2}m^{1/2} = n^{1/2}$.

If t is any constant $t > 0$, then the low order term in R becomes $O(n^{1/2+(1-a)/(2t)}\log^{3/2} n)$.

If however $t = \Theta(\log n)$, the low order term in R becomes $O(\sqrt{n}\log^{3/2} n)$, since d is a constant.

From either term we can shave off a $\log n$ term if instead of sorting we find m_i, M_i by applying the algorithm recursively. The probabilistic claims at every iteration fail with probability at most $1/n^{\rho}$, for any $\rho > 0$. We have constant claims per iteration and a maximum of $\log n$ iterations, therefore the probability of failure for the algorithm is $O(\log n/n^{\rho}) = 1/n^{\theta}$ for some $\theta > 0$. Therefore the comparison bound claim holds with high probability.

■

4 Conclusion

We presented an oversampling-based approach to the bootstrapping method by Floyd and Rivest for the selection problem that results in an algorithm that performs $n + C + O(n^{1/2}\log^{3/2} n)$ comparisons with high probability. Although the algorithm is based on standard techniques such as those developed in [7] and [16], the main contribution of this work is the careful analysis and fine-tuning of the performance of the algorithm to achieve the stated running-time bounds with high probability.

References

[1] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18:155-193, 1979.

[2] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7:448-461, 1973.

[3] B. Bollobás. *Random Graphs*. Academic Press, New York, 1984.

[4] W. Cunto and J. I. Munro. Average case selection. In *Proceedings of the 16-th Annual ACM Symposium on Theory of Computing*, pp. 369-375, 1984.

[5] D. Dor and U. Zwick. Selecting the median. *SIAM Journal on Computing*, 28(5):1722-1758, 1999.

[6] D. Dor and U. Zwick. Median selection requires $(2 + \epsilon)n$ comparisons. *SIAM Journal on Discrete Mathematics*, 14(3):312-325, 2001.

[7] R. W. Floyd and R. L. Rivest. Expected time bounds for selection. *Communications of the ACM*, 18:165-172, 1975.

[8] A. V. Gerbessiotis and L. G. Valiant. Direct Bulk-Synchronous Parallel Algorithms. *Journal of Parallel and Distributed Computing*, 22:251-267, 1994.

[9] A. V. Gerbessiotis and C. J. Siniolakis. Deterministic sorting and randomized median finding on the BSP Model. In *Proceedings of the 8-th ACM Symposium on Parallel Algorithms and Architectures*, June 1996.

[10] A. V. Gerbessiotis and C. J. Siniolakis. Concurrent Heaps on the BSP model. Technical Report PRG-TR-14-96, Computing Laboratory, Oxford University, July 1996.

[11] L. Hyafil. Bounds for selection. *SIAM Journal of Computing*, 5:109-114, 1976.

[12] D. G. Kirkpatrick. A unified lower bound for selection and set partitioning problems. *Journal of the ACM*, 28:150-165, 1981.

[13] D. E. Knuth. *The Art of Computer Programming. Volume I: Fundamental Algorithms.* Addison-Wesley, Reading, 1969.

[14] D. E. Knuth. *The Art of Computer Programming. Volume III: Sorting and Searching.* Addison-Wesley, Reading, 1973.

[15] J. I. Munro and P. V. Poblete. A lower bound for determining the median. Research Report CS-82-21, University of Waterloo, 1982.

[16] H. J. Reif and L. G. Valiant. A logarithmic time sort for linear size networks. *Journal of the ACM,* 34:60-76, January 1987.

[17] A. Schonhage, M. Paterson, and N. Pippenger. Finding the median. *Journal of Computer and System Sciences*, 13:184-199, 1976.

A Proof of Proposition 1

Proof We first observe that the combinatorial identity [13],

$$\sum_{n-u \le v \le r-m} \binom{u+v}{n} \binom{r-v}{m} = \binom{r+u+1}{m+n+1},$$

implies that

$$\sum_{j=2c\lambda s-\mu t/2-1}^{N-2(1-c)\lambda s-\mu t/2} \binom{j}{2c\lambda s-\mu t/2-1} \binom{N-j-1}{2(1-c)\lambda s+\mu t/2-1} = \binom{N}{2\lambda s-1}.$$

Let $B = cN - f(N) - 1$. We use a combinatorial interpretation argument, as in [8], to argue that the sum $\sum_{j \le B} p(j)$ for $j \le B$ is the number of ways we can choose $2\lambda s - 1$ objects so that among them, the object with the $(2c\lambda s - \mu t/2)$-th highest label is labeled at most B.

In order to establish an upper bound for $\sum_{j \le B} p(j)$ we approximate $p(j)$ by a binomial term and use bounds on the right tail of the binomial distribution. Let $q = (2\lambda s - 1)/N$. The probability that in N Bernoulli trials, each with a probability q of success we get $2c\lambda s - \mu t/2 - 1$ successes among the first j, a success at the $(j+1)$-st trial, and $2(1-c)\lambda s + \mu t/2 - 1$ successes in the remainder is

$$\begin{aligned} r(j) &= \binom{j}{2c\lambda s-\mu t/2-1} q^{2c\lambda s-\frac{\mu t}{2}-1}(1-q)^{j-2c\lambda s+\frac{\mu t}{2}+1} \\ &\quad q \binom{N-j-1}{2(1-c)\lambda s+\mu t/2-1} q^{2(1-c)\lambda s+\frac{\mu t}{2}-1}(1-q)^{N-j-2(1-c)\lambda s-\frac{\mu t}{2}}. \end{aligned}$$

Let A be the ratio of $r(j)$ over $p(j)$. The following then holds.

$$A = \frac{r(j)}{p(j)} = \binom{N}{2\lambda s-1} q^{2\lambda s-1}(1-q)^{N-2\lambda s+1}.$$

The following inequality is shown in [3].

$$e^{-\frac{1}{6c}} \frac{1}{\sqrt{2\pi abd}} \left(a^a b^b\right)^{-d} \le \binom{d}{c}.$$

By substituting for $d = N$, $c = 2\lambda s - 1$, $a = q = (2\lambda s - 1)/N$, and $b = 1 - a$ we get the following result.

$$\frac{e^{-\frac{1}{6(2\lambda s-1)}} q^{-2\lambda s+1} (1-q)^{-N+2\lambda s-1}}{\sqrt{2\pi(2\lambda s-1)(1-(2\lambda s-1)/N)}} \le \binom{N}{2\lambda s-1}.$$

It is then immediate that the following holds.

$$A \ \ge \ \frac{e^{-\frac{1}{6(2\lambda s-1)}}}{\sqrt{2\pi(2\lambda s-1)(1-(2\lambda s-1)/N)}}. \tag{11}$$

Thus, we obtain the following result.

$$\sum_{j \le B} p(j) \;\; \le \;\; \frac{q}{A} \sum_{j \le B} \binom{j}{2c\lambda s - \mu t/2 - 1} q^{2c\lambda s - \mu t/2 - 1} (1-q)^{j - 2c\lambda s + \mu t/2 + 1}. \qquad (12)$$

Our goal is to establish an upper bound for the term below.

$$\sum_{j \le B} \binom{j}{2c\lambda s - \mu t/2 - 1} q^{2c\lambda s - \mu t/2 - 1} (1-q)^{j - 2c\lambda s + \mu t/2 + 1}.$$

We employ generating functions. The term

$$\binom{j}{2c\lambda s - \mu t/2 - 1} q^{2c\lambda s - \mu t/2 - 1} (1-q)^{j - 2c\lambda s + \mu t/2 + 1}$$

is the coefficient of $x^{2c\lambda s - \mu t/2 - 1}$ in the polynomial $((1-q)+qx)^j$. Therefore, the sum over all those terms with $j \le B$ is the coefficient of $x^{2c\lambda s - \mu t/2 - 1}$ in the following sum.

$$\sum_{j \le B} ((1-q)+qx)^j \;\; = \;\; \frac{((1-q)+qx)^{B+1}}{-q(1-x)} + \frac{1}{q(1-x)}.$$

The coefficient of $x^{2c\lambda s - \mu t/2 - 1}$ in this expression is the following one.

$$\frac{1}{q} - \frac{1}{q} \sum_{i=0}^{2c\lambda s - \mu t/2 - 1} \binom{B+1}{i} q^i (1-q)^{B+1-i} = \frac{1}{q} \sum_{i=2c\lambda s - \mu t/2}^{B+1} \binom{B+1}{i} q^i (1-q)^{B+1-i}.$$

The sum in the term above gives the probability of having at least $2c\lambda s - \mu t/2$ successes in $B+1$ independent Bernoulli trials with probability of success q. Chernoff's upper bound [1] on the probability of having at least $(1+\beta)(B+1)q$ successes in such an experiment gives an upper bound of $e^{-(\beta^2(B+1)q)/3}$. We set $2c\lambda s - \mu t/2 = (1+\beta)(B+1)q$ in this bound and get $e^{-(2c\lambda s - \mu t/2 - (B+1)q)^2/(3(B+1)q)}$. We choose the function $f(N)$ so that $f(N) = (N\ \mu t)/(2\ \lambda s)$. The following lower bound for $2c\lambda s - \mu t/2 - (B+1)q$ can be obtained.

$$2c\lambda s - \mu t/2 - (B+1)q \;\; \ge \;\; \mu t/2 - (\mu t)/(2\lambda s) + c \ge \mu t/2 - (\mu t)/(2\lambda s).$$

An upper bound for $(B+1)q$ is derived similarly.

$$(B+1)q \;\; \le \;\; (2c\lambda s - \mu t) + (\mu t)/(2\ \lambda s) - c \le (2c\lambda s - \mu t) + (\mu t)/(2\ \lambda s).$$

We thus get for the sum of interest.

$$\sum_{j\le B}\binom{j}{2c\lambda s-\mu t/2-1}q^{2c\lambda s-\frac{\mu t}{2}-1}(1-q)^{j-2c\lambda s+\frac{\mu t}{2}+1} = \frac{1}{q}\sum_{i=2c\lambda s-\mu t/2}^{B+1}\binom{B+1}{i}q^i(1-q)^{B+1-i}$$
$$\le e^{-(2c\lambda s-\mu t/2-(B+1)q)^2/(3(B+1)q)}. \qquad (13)$$

The desired bound follows from equations (11), (12) and (13) and is given by the following expression.

$$\sum_{j\le B}p(j) \le \sqrt{2\pi(2\lambda s-1)}\,e^{\frac{1}{6(2\lambda s-1)}}\,e^{-\frac{(2c\lambda s-\mu t/2-(B+1)q)^2}{3(B+1)q}}.$$

From the lower bound of $2c\lambda s-\mu t/2-(B+1)q$ derived earlier and the corresponding upper bound for $(B+1)q$ we get that, since $c\le 1$, and for $\mu t\ge 6\ge 4c+2c\ge 4c+\mu t/(2\lambda s)$,

$$e^{-(2c\lambda s-\mu t/2-(B+1)q)^2/(3(B+1)q)}\le e^{-(1/3)(\mu t/2)^2/(2c\lambda s)}.$$

Substituting for $\mu t = 2\sqrt{3\rho\, 2c\,\lambda s\, \log n}$ and noting that $\sqrt{2\pi(2\lambda s-1)}\ e^{\frac{1}{6(2\lambda s-1)}} < n$, gives the desired result. ■