One way to abstract the median-finding algorithm is as follows:

— Pick \( d \) such that the rank of \( d \) in the original set \( S \) is between \( n/2 - 2n^{2/3} \) and \( n/2 \). Let \( \ell_d \) be the rank of \( d \) in \( S \).

— Pick \( u \) such that the rank of \( u \) in the original set \( S \) is between \( n/2 \) and \( n/2 + 2n^{2/3} \).

— Let \( C \) be the set of elements in \( S \) between \( d \) and \( u \) and sort \( C \). Output the \((n/2 - \ell_d + 1)\)th element in \( C \).

Note that given \( s \), we can compute the rank of \( s \) in \( S \) in linear time. Therefore, we can always test in linear time whenever the ranks of \( d \) and \( u \) lie in the given range. Moreover, whenever the ranks of \( d \) and \( u \) do lie in the given range, it is the case that:

- \(|C| \leq 4n^{2/3} \), and thus we can sort \( C \) in linear time;
- the median lies in \( C \), so the algorithm must output the median correctly.

It is then clear that there are four ways in which the algorithm could fail:

- rank \( d \) is greater than \( n/2 \), namely
  \[ \mathcal{E}_1 : \quad |\{r \in R \mid r \leq m\}| < \frac{1}{2} n^{3/4} - \sqrt{n} \]

- rank \( d \) is less than \( n/2 - 2n^{2/3} \) (\( \mathcal{E}_{3,2} \) in the text)

- rank \( u \) is less than \( n/2 \) (\( \mathcal{E}_2 \) in the text)

- rank \( u \) is greater than \( n/2 + 2n^{2/3} \) (\( \mathcal{E}_{3,1} \) in the text)

where \( R \) is a set of \( n^{3/4} \) elements of \( S \) chosen u.a.r. with replacement. We can show that each of these events occurs with probability at most \( \frac{1}{4} n^{-1/4} \).
In section, we studied an algorithm related to picking \(d\) in the above exposition. Specifically, we are given an unsorted list \(S\), comprising distinct unknown numbers \(\alpha_1 < \alpha_2 < \cdots < \alpha_n\) in some unknown order. Our goal is to output in linear time a number \(d\) satisfying \(\alpha_{n/2-o(n)} < d < \alpha_{n/2}\). In particular, \(d\) is a “good” lower bound for the median. The algorithm we will analyze is as follows, parameterized by a constant \(\gamma \in (0.5, 0.9)\):

— Pick a random set \(R\) of size \(n^\gamma\) u.a.r. from \(S\) with replacement.

— Sort the set \(R\) and output \(d\), the \((1/2 n^\gamma - n^{2\gamma/2})\)th smallest element in \(R\).

We want to compute the probability that \(d\) lies between \(\alpha_{n/2-2n^{1-\gamma/2}}\) and \(\alpha_{n/2}\) (that is, \(d\) has rank between \(n/2 - 2n^{1-\gamma/2} = n/2 - o(n)\) and \(n/2\) in the list \(S\)). We do this by computing \(\Pr[d > \alpha_{n/2}]\) and \(\Pr[d < \alpha_{n/2-2n^{1-\gamma/2}}]\).

(a) Let \(X_1, \ldots, X_{n^\gamma}\) be independent 0-1 r.v.’s such that \(\Pr[X_1 = 1] = p\). Let \(Y = X_1 + \cdots + X_{n^\gamma}\), so \(Y\) is a binomial r.v. with mean \(pn^\gamma\). Using Chebyshev’s inequality, show that:

\[
\Pr\left[|Y - E[Y]| \geq n^{\gamma/2}\right] \leq \frac{1}{4}
\]

(b) Using (a), show that \(\Pr[d \geq \alpha_{n/2}] \leq \frac{1}{4}\).

[HINT: Consider the r.v. \(Y_1\) which is the number of elements in \(R\) less than \(\alpha_{n/2}\) and observe that \(\Pr[d \geq \alpha_{n/2}] = \Pr[Y_1 < \frac{n^\gamma}{2} - n^{\gamma/2}]\).]

(c) Using (a), show that \(\Pr[d < \alpha_{n/2-2n^{1-\gamma/2}}] \leq \frac{1}{4}\).

[HINT: Consider the r.v. \(Y_2\) which is the number of elements in \(R\) less than \(\alpha_{n/2-2n^{1-\gamma/2}}\) and observe that \(\Pr[d < \alpha_{n/2-2n^{1-\gamma/2}}] = \Pr[Y_2 \geq \frac{n^\gamma}{2} - n^{\gamma/2}]\). You will also need to show that \(E[Y_2] = \frac{1}{2} n^{\gamma} - 2n^{\gamma/2}\).]

(d) Show that with probability at least \(1/2\), the algorithm outputs a number between \(\alpha_{n/2-o(n)}\) and \(\alpha_{n/2}\).

(e) Show that the running time of the algorithm is \(O(n^\gamma \log n)\). (Assume we can do comparisons and sample an element from \(S\) in constant time.)

(f) Show how we may modify the algorithm to always output a number between \(\alpha_{n/2-o(n)}\) and \(\alpha_{n/2}\) in expected linear time.

Can we improve the analysis of the algorithm using Chernoff bound?

(g) Use the Chernoff bound to obtain an upper bound on \(\Pr[d > \alpha_{n/2}]\).

[TIP: The relevant variant of Chernoff bound is the one which says \(\Pr[X \geq (1 + \delta)\mu] \leq e^{-\mu\delta^2/3}\), where \(0 < \delta \leq 1\).]

(h) Suppose we modify the algorithm to output the \((1/2 n^{\gamma} - n^{2\gamma/3})\)th smallest element in \(R\) and call that number \(d'\). Now, use Chebyshev’s inequality and Chernoff bound to obtain upper bounds on \(\Pr[d' > \alpha_{n/2}]\).