CS 6550: Randomized Algorithms **Spring 2019**

Lecture 2: Median Finding

January 10, 2019

Lecturer: Eric Vigoda *Scribes: Daniel Hathcock, Shyamal Patel*

2.1 Classical Median Finding Algorithms

Given an unsorted list of n numbers $S = [S_1, \ldots, S_n]$, find the median (n is assumed to be odd for convenience). There are many algorithms to do this:

- Easy: **sort** in $O(n \log n)$ time, output middle element.
- Non-trivial: **median of medians** algorithm, deterministically finds median in $O(n)$ time via divide and conquer, originally by Blum et. al [2].
- Trivial (randomized): **quickselect** algorithm finds median in $O(n)$ expected time
- Non-trivial (randomized): the main algorithm presented here finds the median in $O(n)$ with high probability (whp).

Definition 2.1. *An event A_n which depends on some parameter n occurs **with high probability (whp)** if*

$$\mathbb{P}(A_n) \geq 1 - \frac{1}{n^c}$$

for some $c > 0$. Or sometimes, alternatively, if

$$\mathbb{P}(A_n) \geq 1 - o(1)$$

The randomized **quickselect** approach to finding the median of S is to choose some "good" pivot p, then partition S into $S_{<p}$, $S_{=p}$, and $S_{>p}$. Then recurse, using the sizes of parts to determine which contains the median.

What does it mean to have a good pivot? $|S_{<p}| \leq \frac{3}{4}n$ and $|S_{>p}| \leq \frac{3}{4}n$. If this is the case, then the running time has the recurrence relation $T(n) \leq T(\frac{3}{4}n) + O(n)$ which is $O(n)$ by the master theorem. (note that $\frac{3}{4}$ is arbitrary. Any constant larger than $\frac{1}{2}$ and less than 1 is fine).

To find a good pivot, choose randomly. Half of the elements are good pivots, so in expectation, we will have to choose two pivots. Since the pivot is good in expectation after only $O(1)$ trials, then the algorithms finishes in expected $O(n)$ time. However, if we want to get $O(n)$ time with high probability instead, then we need a different algorithm.

2.2 Find Median With High Probability

Idea: Suppose m is the median. Choose some ℓ and u such that $\ell \leq m \leq u$, and both ℓ and u are "close" to m. Consider the set

$$C = \{x \in S \mid \ell \leq x \leq u\}, \qquad \text{with } |C| = o\left(\frac{n}{\log n}\right)$$

C contains the median by our assumption on ℓ and u, and it is small enough to sort efficiently. So, we sort C and find the $(\frac{n+1}{2} - |S_{<\ell}|)^{th}$ smallest element, which is the median of S.

To find ℓ and u, choose a random subset $R \subseteq S$ of size $n^{3/4}$, and sort it. It is hard to sample a true subset, so instead choose R by sampling $n^{3/4}$ elements randomly with replacement, creating a multiset. We let

$$\ell = \left(\frac{n^{3/4}}{2} - \sqrt{n}\right)^{\text{th}} \text{ smallest element} \qquad \text{and} \qquad u = \left(\frac{n^{3/4}}{2} + \sqrt{n}\right)^{\text{th}} \text{ smallest element}$$

using $\sqrt{n}$ intuitively as the standard deviation (and also because this is what makes the analysis work out nicely). This is the entire algorithm, with the condition that the algorithm returns failure if the randomly chosen ℓ and u are not good enough (either the median is not between them, or they are too far apart):

Algorithm 1: Randomized Find Median with high probability

input : List S of n integers, n odd
output: The median of S, or FAIL

1 $R \leftarrow$ choose $n^{3/4}$ elements from S (u.a.r. and with replacement);
2 sort R;
3 $\ell \leftarrow \left(\frac{n^{3/4}}{2} - \sqrt{n}\right)^{\text{th}}$ smallest element of R;
4 $u \leftarrow \left(\frac{n^{3/4}}{2} + \sqrt{n}\right)^{\text{th}}$ smallest element of R;
5 $C \leftarrow \{x \in S \mid \ell \le x \le u\}$, $S_{<\ell} \leftarrow \{x \in S \mid x < \ell\}$, $S_{>u} \leftarrow \{x \in S \mid x > u\}$;
6 **if** $|S_{<\ell}| \ge \frac{n}{2}$ or $|S_{>u}| \ge \frac{n}{2}$ **then**
7 output FAIL;
8 **if** $|C| > 4n^{3/4}$ **then**
9 output FAIL;
10 **else**
11 sort C;
12 output $\left(\frac{n+1}{2} - |S_{<\ell}|\right)^{\text{th}}$ smallest element of C;

Note that the above algorithms works as long as the elements in S are distinct (consider the case where S contains only many copies of 1, then C will always have size n) If S is allowed to have repeat elements, then the algorithm must be slightly modified so that in line 5, the comparisons being done take into account a total order defined on the elements (for example, where ties are broken based on the original index in the list S).

2.3 Analysis

2.3.1 Runtime Analysis

Claim 2.2. *The randomized find median with high probability algorithm runs in $O(n)$ time.*

Proof. Sampling $n^{3/4}$ elements with replacement takes $o(n)$ time. Then, sorting R takes

$$O(n^{3/4} \log n^{3/4}) = o(n) \text{ time}$$

After sorting, finding ℓ and u are both $O(1)$ operations. Partitioning S into C, $S_{<\ell}$, and $S_{>u}$ can easily (naively) be done by passing over S and comparing each element to both ℓ and u. This clearly takes $O(n)$ time. Checking the sizes of each of the sets in the partition is constant time, and assuming FAIL was not outputted, then sorting C takes

$$O(4n^{3/4} \log 4n^{3/4}) = o(n) \text{ time}$$

Finally, finding the median of S from the sorted list C is constant time. So, in total, each of the constant number of steps is $O(n)$, so the entire algorithm takes $O(n)$ time □

There is a lower bound proved on the number of comparisons required for deterministic median finding algorithms in the worst case: $2n + o(n)$ [1]. A single run of the presented randomized algorithm (assuming it succeeds), can be shown to perform better (by a constant factor) if line 5 of the algorithm is implemented slightly more efficiently.

Claim 2.3. *The randomized median finding algorithm performs $\frac{3}{2}n + o(n)$ comparisons in the worst case (assuming success).*

Proof. The only comparisons performed in the algorithm are in the steps which sort R, sort C, and partition S. Assuming success, sorting R and C both take $o(n)$ comparisons.

Partitioning S can be done by passing over S once, comparing each element to ℓ, and only comparing it to u if it is greater than ℓ. This results in

$$n + (n - |S_{<\ell}|) \text{ comparisons}$$

However, notice that

$$|S_{<\ell}| < \frac{n}{2} - 4n^{3/4} \implies |S_{>u}| > \frac{n}{2} \text{ or } |C| > 4n^{3/4} \implies \text{output FAIL}$$

Since we assumed that the algorithm does not output fail, this means that $|S_{<\ell}| \geq \frac{n}{2} - 4n^{3/4} = \frac{n}{2} - o(n)$. So, the total number of comparisons is

$$2o(n) + n + (n - |S_{<\ell}|) \leq 2o(n) + 2n - \frac{n}{2} + o(n) = \frac{3}{2}n + o(n)$$ □

2.3.2 Failure Probability

We will show that the probability of failure $\mathbb{P}(\text{FAIL}) < n^{-1/4}$. Showing this statement in turn implies that we succeed with high probability. To show this we first define the following events:

$$\mathcal{E}_1 := |S_{<\ell}| \geq \frac{n}{2}$$
$$\mathcal{E}_2 := |S_{>u}| \geq \frac{n}{2}$$
$$\mathcal{E}_3 := |C| > 4n^{3/4}$$

We will show that

Claim 2.4. $\mathbb{P}(\mathcal{E}_1), \mathbb{P}(\mathcal{E}_2) < \frac{1}{4}n^{-1/4}$

Claim 2.5. $\mathbb{P}(\mathcal{E}_3) < \frac{1}{2}n^{-1/4}$

Using a union bound argument, we then get that these two claims imply

$$\mathbb{P}(\text{FAIL}) \leq \mathbb{P}(\mathcal{E}_1) + \mathbb{P}(\mathcal{E}_2) + \mathbb{P}(\mathcal{E}_3) < n^{-1/4}$$

as desired.

2.3.2.1 Bernoulli Random Variables and Chebychev's Inequality

Before we proceed with the proofs, we will first recall some facts about Bernoulli random variables. Recall that if $X_i \sim \text{Bernoulli}(p)$ then

$$X_i = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$$
$$\mathbb{E}[X_i] = p$$
$$\text{Var}(X_i) = \mathbb{E}[X_i^2] - (\mathbb{E}[X_i])^2 = p - p^2 = p(1-p)$$

Moreover, if the variables $X_1, X_2, ..., X_s$ are independent and we let $Z = X_1 + ... + X_s$ then

$$\text{Var}(Z) = sp(1-p)$$

Another fact that will be key in proving the two claims is

Theorem 2.6 (Chebychev's Inequality)**.** *For a random variable, X, we have that:*

$$\mathbb{P}\left(\left|X - \mathbb{E}[X]\right| \geq a\right) \leq \frac{Var(X)}{a^2}$$

2.3.2.2 Proofs

We will first show that $\mathbb{P}(\mathcal{E}_1)$ and $\mathbb{P}(\mathcal{E}_2)$ are both small, which implies that the center will likely contain the median.

Proof of Claim 2.4. We will only show $\mathbb{P}(\mathcal{E}_1) < \frac{1}{4}n^{-1/4}$. The other inequality will follow by an analogous argument.

Notice that in order to have that $|S_{<\ell}| \geq \frac{n}{2}$ we must have that $|\{r \in R : r \leq m\}| < \frac{n^{3/4}}{2} - \sqrt{n}$. So now let $R = \{r_1, ..., r_{n^{3/4}}\}$ and define the random variable X_i such that

$$X_i = \begin{cases} 1 & \text{if } r_i \leq m \\ 0 & \text{otherwise} \end{cases}$$

And let $Y = \sum_{i=1}^{n^{3/4}} X_i$. Now notice that the random variables X_i are Bernoulli random variables. As such we can calculate the expectations and variance for each X_i and Y.

$$E[X_i] = \mathbb{P}(X_i = 1) = \frac{\frac{n-1}{2} + 1}{n} = \frac{1}{2} + \frac{1}{2n}$$
$$\mathbb{E}[Y] = n^{3/4}\left(\frac{1}{2} + \frac{1}{2n}\right) = \frac{n^{3/4}}{2} + \frac{1}{2n^{1/4}}$$
$$\text{Var}(Y) = n^{3/4}\left(\frac{1}{2} + \frac{1}{2n}\right)\left(\frac{1}{2} - \frac{1}{2n}\right) = \frac{n^{3/4}}{4} - \frac{1}{4n^{5/4}} < \frac{n^{3/4}}{4}$$

Now by Chebychev's Inequality we have that

$$\mathbb{P}(\mathcal{E}_1) \leq \mathbb{P}(Y < \frac{n^{3/4}}{2} - \sqrt{n}) \leq \mathbb{P}(|Y - E[Y]| > \sqrt{n}) \leq \frac{\text{Var}(Y)}{n} < \frac{n^{3/4}}{4n} = \frac{1}{4n^{1/4}}$$

which completes the proof. □

So now it only remains to show that the center will be small with high probability.

Proof of Claim 2.5. Notice that if $|C| > 4n^{3/4}$ then we must have that either $|\{x \in C : x \geq m\}| > 2n^{3/4}$ or $|\{x \in C : x \leq m\}| > 2n^{3/4}$. We will show that $\mathbb{P}(|\{x \in C : x \geq m\}| > 2n^{3/4}) < \frac{1}{4n^{1/4}}$.

Notice that if we have that $|\{x \in C : x \geq m\}| > 2n^{3/4}$ then we must have that u is one of the $\frac{n}{2} - 2n^{3/4}$ largest elements of S. Hence at least $\frac{n^{3/4}}{2} - \sqrt{n}$ elements of R, namely the elements of R that are bigger than u, are from the $\frac{n}{2} - 2n^{3/4}$ largest elements of S. So now let:

$$X_i = \begin{cases} 1 & \text{if } r_i \geq \frac{n}{2} + 2n^{3/4} \text{ smallest element of } S \\ 0 & \text{otherwise} \end{cases}$$

Then we have that

$$\mathbb{E}[X_i] = \frac{\frac{n}{2} - 2n^{3/4}}{n} = \frac{1}{2} - \frac{2}{n^{1/4}}$$

If we again define $Y = \sum_{i=1}^{n^{3/4}} X_i$ then it follows that

$$\mathbb{E}[Y] = n^{3/4}\left(\frac{1}{2} - \frac{2}{n^{1/4}}\right) = \frac{n^{3/4}}{2} - 2\sqrt{n}$$

$$\mathrm{Var}(Y) = n^{3/4}\left(\frac{1}{2} + \frac{2}{n^{1/4}}\right)\left(\frac{1}{2} - \frac{2}{n^{1/4}}\right) = \frac{n^{3/4}}{4} - 4n^{1/4} < \frac{n^{3/4}}{4}$$

So by Chebychev's Inequality we have that

$$\mathbb{P}(Y \geq \frac{n^{3/4}}{2} - \sqrt{n}) \leq \mathbb{P}\left(\left|Y - \mathbb{E}[Y]\right| \geq \sqrt{n}\right) \leq \frac{\mathrm{Var}(Y)}{n} < \frac{n^{3/4}}{4n} = \frac{1}{4n^{1/4}}$$

By an analogous argument, we have that $\mathbb{P}(|\{x \in C : x \leq m\}| > 2n^{3/4}) < \frac{1}{4n^{1/4}}$. Hence, by another union bound argument, we have that

$$P(\mathcal{E}_3) < \frac{1}{4n^{1/4}} + \frac{1}{4n^{1/4}} = \frac{1}{2n^{1/4}}$$

□

References

[1] S. Bent and J. John. Finding the median requires 2n comparisons. *In Proceedings of the 17th Annual ACM Symposium on Theory of Computing, Providence, Rhode Island*, pages 213216, 1985

[2] M. Blum, R.W. Floyd , V. Pratt , R.L. Rivest , and R.E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, v.7 n.4, pages 448–461, 1973.