CS41 Lab 13

November 30 2021

This week, we'll continue exploring the use of randomization and approximation to solve problems.

1. **Finding the Median.** Given a set $S = \{a_1, \ldots, a_n\}$ of numbers, the *median* of S, denoted $\text{med}(S)$, is the k-th smallest element of S, where $k = \lfloor \frac{n+1}{2} \rfloor$. In this problem, you will analyze a randomized algorithm to output the median. Consider the following algorithm for finding the median:

```
FindMedian(S)
    Return Select(S, ⌊(n+1)/2⌋)

Select(S, k)
    Choose pivot a_i ∈ S at random, uniformly[a]
    Initialize S−, S+ := {}
    for each j ≠ i
        if a_j < a_i add a_j to S−
        if a_j > a_i add a_j to S+
    if |S−| = k − 1 return a_i
    else if |S−| > k − 1
        return Select(S−, k)
    else
        return Select(S+, k − (1 + |S−|))
```

[a] An element is chosen *uniformly* if each element is equally likely to be picked.

- First, show that FindMedian always returns the median.
- Next, analyze the running time of FindMedian when the pivot element is chosen uniformly from S. The following structure will help guide you.

  Say that the algorithm is in **phase** j if there are between $n(3/4)^j$ and $n(3/4)^{j+1}$ elements in the set S being considered. So, for example, we are in phase 0 the first time Select is called (because $n(3/4)^{0+1} \leq |S| \leq n(3/4)^0$).

  Call an element $a_i \in S$ **central** to S if (i) at least $|S|/4$ of the elements of S are less than a_i and (ii) at least $|S|/4$ elements of S are greater than a_i.

  (a) Show that there are $|S|/2$ central elements.

  (b) Show that if the pivot element is central, the phase ends i.e., the next recursive call that gets made will be in a different phase.

  (c) Give an upper bound on the expected number of recursive calls to Select before a round ends.

  (d) Give an upper bound on the running time of each Select call, not including recursive calls.

  (e) Give an upper bound on the number of phases that are run before FindMedian terminates.

(f) Give an upper bound on the expected runtime of FindMedian when the pivot is chosen uniformly.

2. **Chromatic Number.** Consider the optimization problem ChromaticNumber, defined as follows. Given a graph $G = (V, E)$ as input, determine the smallest number k such that it is possible to k-color the graph.

   (a) Prove that ChromaticNumber is NP-hard.

   (b) Prove that there is no efficient $\frac{4}{3}$-approximation to ChromaticNumber unless P = NP.