Lecture 38: Median Finding

10:00 AM, Dec 4, 2019

Contents

1 Median Finding

2 Analysis of RandSelect

3 Next Steps

1 Median Finding

We did a recap on median finding from the last class: the SELECT algorithm (Blum, Floyd, Pratt, Rivest, Tarjann, 1973)

Now there’s a newer, simpler algorithm, RandSelect. We discussed its benefits, drawbacks, and challenges, some of which are covered here.

2 Analysis of RandSelect

How many iterations of choosing a new x might we need to go through at most n? We’re doing a certain amount of work at each step - this ends up being quadratic at worst case. But how bad a day does it have to be in order to be quadratic time? You would have to choose the highest number in the set every time. Pretty unlikely.

Half the time, the random number will be in the middle half of the set of numbers. That fact is enough to make a conclusion that this algorithm runs in "expected linear time". On average, the size of the larger partition is at most 3/4 the size of the overall set.

How do you get the random number to split the set? In a functional programming language, all functions return the same output for a given input (principle of functional programming). How can rand output different numbers each time, and then test for an expected output?

3 Next Steps

We began summarizing what we’ve learned throughout the semester, and what our next steps are in our future CS classes.

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS 17 document by filling out the anonymous feedback form: [http://cs.brown.edu/courses/csci0170/feedback](http://cs.brown.edu/courses/csci0170/feedback).