COMP160: Algorithms, **Homework 4**

- To obtain full credit, you must justify your answers. When describing an algorithm, do not forget to **state any assumptions that you make**, analyze its running time and explain why the algorithm is correct.
- Although not specifically stated, you can assume that we look for algorithms that are as fast as possible, and bounds as tight as possible.
- You may discuss these problems with others, but remember to write the answers on your own. In case of doubt, cite any source you used to do the assignment.
- Remember to submit each question in a separate page.

1. We consider a modification of the QuickSort algorithm presented in class. Normally, we pick a pivot at random and use it as pivot. Say that we instead we insert a step to calculate the median-of-medians (say, using groups of 5) and use that as pivot instead. In other words, ModifiedQuickSort is:

   1. Group the n elements in the array in groups of 5. Compute the median for each group (using Selection)
   2. Let m be the median-of-medians
   3. Partition input array around the median-of-medians. Let i be the position of the median-of-medians in the reordered array
   4. Execute ModifiedQuickSort on the two subarrays indicated by Partition (that is, from 0 to $i-1$ and from $i+1$ to n).

   (a) After step 3, what are the possible values for i?

   (b) Write the recursion for the runtime of the algorithm, incorporating IRVs where needed. Explain briefly why your recursion is correct.

   (c) What is the runtime of the algorithm?

   Note: Doing a precise calculation will be very difficult. Instead, take step back and think: how does the expression look like? Can you give a bound using some other argument? Just justify your answer with a few sentences. Make sure to specify if you mean worst case or expected worst-case.

2. Both Quicksort and RandSelect start by choosing a (potentially random) pivot. The algorithm then partitions the array around that pivot, creating a "smaller" group and a "larger" group, and finally recurses in some way using those groups.

   (a) RandSelect has an extra step after partitioning and before recursing; describe that step. What is the runtime of that step alone?

   (b) Both Quicksort and RandSelect use recursion to solve the corresponding problem. What is the main difference between the recursive calls of both problems?

©2020 Tufts University

(c) Follow-up on the previous question: what is the impact of this difference in the formulas for expected runtime? Give the formula for both algorithms (as a sum of expectancies of some IRVs). What is the main difference between the two formulas?

(d) This difference in the formulas may look minor, but it is significant enough to result in a difference of runtimes (linear vs. $\Theta(n \log n)$). Try solving the recurrence relation for QUICKSORT using substitution and the guess $E[T(n)] \leq a \cdot n$. Show that this fails.

3. The end of block 1 is getting closer. Let's practice the kind of questions that could come in an exam:

Oh, no! You are about to introduce your significant other to your parents, when they decide to show all of your baby embarrassing pictures. Hurry! You have to destroy them! Fortunately, they left you alone with them armed with two tools: scissors and a small image incinerator. The incinerator is small can can only fit a picture of unit area (regardless of the shape).

You consider three strategies for destroying a large picture:

Prune and incinerate Use the scissors to cut off a portion of unit area from the large picture. Incinerate the small piece, and repeat that process until the reminder of the picture also has unit area or less (in which case you also incinerate both of them).

Divide and divide again If the picture fits in the incinerator, then destroy it. Otherwise, use the scissors to cut the picture into two pieces of the same area, and recursively destroy those images.

Stack them up As before, but now we cut the picture into two identical pieces[1], stack them and treat them as one. Repeat until all pictures in the stack have unit size (and incinerate them one by one).

Enough chatter. Your parents are about to return! Quickly! Answer some questions:

(a) Because the scissors are noisy, you want to use them as little as possible. For each of the above methods write a recursive formula that denotes the number of cuts that will be needed to destroy as a function of the area of the picture (note that we do not care about the number of times that you will use the incinerator). Justify each formula with a short sentence. If you get a recurrence, find an asymptotic bound as well. Ideally, you can find matching upper and lower bounds, but if you can't just get as close as possible. You can use whatever method you want for solving the recurrence, but use a different method each time (say, solve one with substitution, another with master method, and so on). If it helps, you can assume that the area of the image is a power of two.

Challenge (not needed for full credit) try to solve the *divide and divide again* recurrence with substitution. It is possible to get a tight bound, but you will need a few tricks that are not taught during lectures. You are free to ask TAs for help, but

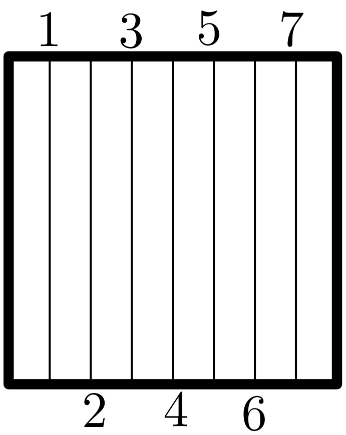
[1] Fun fact: this is always possible. You can try to prove the statement, but for this assignment you need not do so

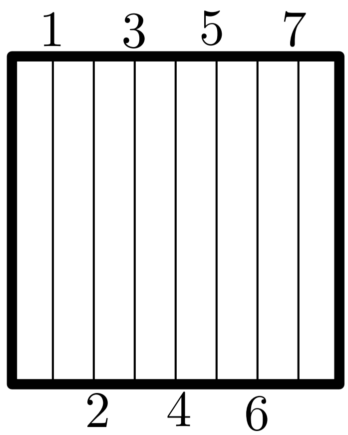
©2020 Tufts University

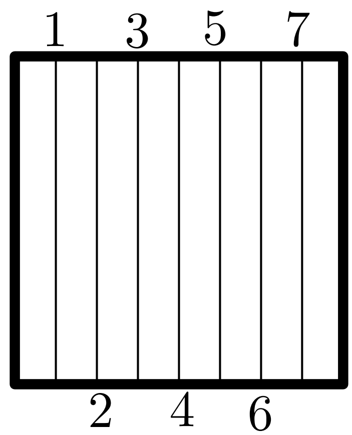
we emphasize that this lands slightly outside the scope of the course. Remember you can use a different technique if you are stuck.

(b) Which of the three strategies is the best one? and the worst one? Why?

Hint: as usual, when facing these type of questions. It is great to solve the problem for small values of n. Below we put three copies of a square of area 8 (and 7 numbered lines that split the area into unit strips). Print this sheet and apply the different algorithm to each of the squares. In which order did you cut the lines with each strategy? Can you generalize this to squares of area n?







©2020 Tufts University