BU CLA CS 530: Analysis of Algorithms

Spring 1996

Problem Set 1

Due Thursday 96.01.25, at 11:00 am, in class. Late submissions not accepted.

---

1. (Review exercise.) Consider the following divide-and-conquer algorithm to find the minimum element in an array $A[1..n]$ of $n \geqslant 1$ integers. The main body of the algorithm makes a single call to the recursive procedure MINIMUM$(A, 1, n)$. The parameters p and r range over $\{1, \ldots, n\}$.

```
MINIMUM(A, p, r)
if r − p ⩽ 1
then return min{A[p], A[r]}
else x ← MINIMUM(A, p, ⌊(p + r)/2⌋)
     y ← MINIMUM(A, 1 + ⌊(p + r)/2⌋, r)
     return min{x, y}
```

   a. Write down a recurrence for each of the best-case, the average-case, and the worst-case, number of comparisons (between elements in the array A) used by MINIMUM$(A, 1, n)$.

   b. Solve the recurrences of part a. (You can assume that n is a power of 2.)

   c. Which is a better algorithm: the recursive (divide-and-conquer) algorithm given here, or the iterative algorithm at the top of page 186 in [CLR]?

2. Exercise 8.3-4, page 163 in [CLR]. (This exercise gives a randomized quicksort algorithm different from RANDOMIZED-QUICKSORT. This is the first approach mentioned in [CLR], page 161, and the requirement is that the input array is randomly permuted in such a way that *every permutation is equally likely.*)

3. Problem 8-5, page 170 in [CLR]. Do parts a and b only, omit parts c and d.

Created: 96.01.18
Modified: 96.01.22