Algorithms Lab 5

(Selection, divide-and-conquer)
Laura Toma, csci2200, Bowdoin College

Review

Topics covered since previous lab:

- selection
- divide-and-conquer: more examples

In lab exercises: Collaboration level : 0

1. (CLRS 9.3-1) In the algorithm SELECT, the input elements are divided into groups of 5. Will the SELECT algorithm work in linear time if they are divided into groups of 7? Argue that SELECT does not run in linear time of groups of 3 or 4 are used.

2. (CLRS 9.3-3) Show how QuickSort can be made to run in $O(n \lg n)$ time in the worst case, assuming that all elements are distinct.

3. Let A be a list of n (not necessarily distinct) integers. Describe an $O(n)$-algorithm to test whether any item occurs more than $\lceil n/2 \rceil$ times in A. Your algorithm should use $O(1)$ additional space. A general solution should not make any additional assumptions about the integers.

4. (CLRS 2-4) Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A.

   (a) List the inversions of the array $< 2, 3, 8, 6, 1 >$.

   (b) What array with elements from the set $\{1, 2, ..., n\}$ has the most inversions? How many does it have?

   (c) Give an algorihm that determines the number of inversions in an array in $O(n^2)$ time.

   (d) Give an algorihm that determines the number of inversions in an array in $O(n \lg n)$ time worst-case (Hint: modify merge sort).

Homework problems: Collaboration level : 1

1. (GT C-4.23, CLRS 9.3-7) Given an unordered sequence S of n elements (for simplicity, assume items are integers or real numbers), describe an efficient method for finding the $\lceil \sqrt{n} \rceil$ elements whose values are closest to (the value of) the median of S. What is the running time of your method? Aim for linear time.

2. (adapted form GT C-4.27, CLRS 9.3-6) Given an unsorted sequence S of n elements, and an integer k, we want to find $O(k)$ elements that have rank $\lceil n/k \rceil$, $2\lceil n/k \rceil$, $3\lceil n/k \rceil$, and so on.

   (a) Describe the "naive" algorithm that works by repeated selection, and analyze its running time function of n and k.

   (b) Describe an improved algorithm that runs in $O(n \lg k)$ time. You may assume that k is a power of 2.

3. (CLRS 9-3.9) Professor Olay is consulting for an oil company, which is planning a large pipeline running east to west through an oil field of n wells.The company wants to connect a spur pipeline from each well directly to the main pipeline along a shortest route (either north or south), as shown in textbook CLRS figure 9.2. Given the x- and y-coordinates of the wells, show how the professor should pick the optimal location of the main pipeline, which would be the one that minimizes the total length of the spurs. Show how to determine the optimal location in linear time. Hint: Assume professor Olay is a computer science major and she loves algorithms!

4. Suppose we are given an array $A[1..n]$ with the special property that $A[1] \geq A[2]$ and $A[n-1] \leq A[n]$. We say that an element $A[x]$ is a *local minimum* if it is less or equal to both its neighbors, or more formally, if $A[x-1] \geq A[x]$ and $A[x \;\leq A[x+1]$. For example. there are six local minima in the following array:

$$A = [9, 7, 7, 2, 1, 3, 7, 5, 4, 7, 3, 3, 4, 8, 6, 9]$$

   .

   We can obviously find a a local minimum in $O(n)$ time by scanning through the array. Describe and analyze an algorithm that finds a local minimum in $O(\lg n)$ time. (*Hint: with the given boundary conditions, the array must have at least one local minimum. Why?*)