Week 6: Lab

Collaboration level 0 (no restrictions). Open notes.

Topics: asymptotic analysis, comparison-based sorting, sorting lower bound, linear-time sorting, heaps, selection.

As you solve each problem below, write down your solution and ask for feedback on the writing. One of the skills you'll develop in this class is how to communicate technical material clearly. Your solutions need to be clear and easily human-readable.

1. Let A be a list of n (not necessarily distinct) integers. Describe an $O(n)$-algorithm to test whether any item occurs more than $\lceil n/4 \rceil$ times in A.

   We expect: (1) pseudocode and an English description of your algorithm; (2) why is it correct; (3) analysis of its running time.

2. (GT C-2.31) Develop an algorithm that computes the kth smallest element in a set of n distinct integers in $O(n + k \lg n)$ time.

   We expect: (1) pseudocode and an English description of your algorithm; (2) analysis of its running time.

3. (C-4.22) Let A and B be two sequences of n integers each. Given an integer x, describe an $O(n \lg n)$ algorithm for determining if there is an integer a in A and an integer b in B such that $x = a + b$.

   We expect: (1) pseudocode and an English description of your algorithm; (2) analysis of its running time.

   (b) Generalize to 3-sum: Find if there exist 3 elements in the array whose sum is k, or report that no such subset exists. Analyze running time.

   We expect: (1) pseudocode and an English description of your algorithm; (2) analysis of its running time.

4. (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 (do not assume k to be a constant).

(b) Describe an improved algorithm that runs in $O(n \lg k)$ time. You may assume that k is a power of 2. After you describe it, argue why its running time is $O(n \lg k)$.

We expect: pseudocode and an English description of your algorithm, and analysis of its running time.

5. 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:

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

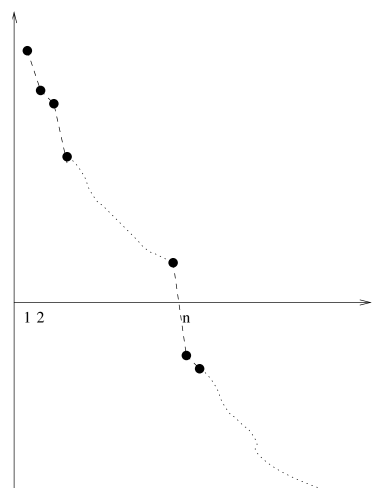
.

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?*)

We expect: (1) pseudocode and an English description of your algorithm; (2) why is it correct; (3) analysis of its running time.

Optional problems

1. Consider a *monotonically decreasing* function $f : N \rightarrow Z$ (that is, a function defined on the natural numbers taking integer values, such that $f(i) > f(i+1)$). Assuming we can evaluate f at any i in constant time, we want to find $n = \min\{i \in N | f(i) \leq 0\}$ (that is, we want to find the value where f becomes negative).



   We can obviously solve the problem in $O(n)$ time by evaluating $f(1), f(2), f(3), \ldots f(n)$. Describe an $O(\log n)$ algorithm.

   (*Hint*: Evaluate f on $O(\log n)$ carefully chosen values between 1 and $2n$ - but remember that you do not know n initially).

2. You accepted an internship as consultant for a green company, which is planning a large underground water pipeline running east to west through a field of n water wells.The company wants to connect a spur pipeline from each well directly to the main pipeline along a shortest route (that is, either north or south, perpendicular to the main pipe). The the x- and y-coordinates of the wells are given. For a certain location of the main pipeline, the total length of the spur for that location is defined as the sum of the lengths of the spurs of each well to the main pipeline. The goal is to pick the optimal location of the main pipeline that minimizes the total length of the spur.

   Describe how you will pick the optimal location of the main pipeline, and argue why it's optimal. Describe your argument for the case n is odd; then extend it to when n is even.