

CS4321 Homework 5

Due Tuesday, Nov. 27 at the beginning of class.

(62 points)

1. (6 points) Problem 2.3-1, P.36. (You can also follow the style in my slides but now use the adhoc sorting only when the sub-array size is 1).
2. (6 points) Problem 7.1-1, P.148.
3. (10 points) Problem 9.3-3, P.192. Write the pseudo code of your algorithm and justify the cost is $O(n \log n)$.
4. (10 points, optional) Problem 9.3-8, P.193.
5. (10 points) Let $A[1..n]$ be a sorted array of distinct integers, some of which may be negative. Give an algorithm that can find an index i such that $1 \leq i \leq n$ and $A[i] = i$, provided such an index exists. Your algorithm should take a time in $O(\log n)$ in worst case and you need to justify it.
6. (10 points) Let $A[1..n]$ be an array of n integers. An integer is a *majority element* in A if it appears strictly more than $n/2$ times. Give an algorithm that can decide whether an array $A[1..n]$ contains a majority element, and if so find it. Your algorithm must run in linear time in the worst case.
Hint: Think about the relation between the median and the majority element.
7. (10 points) Rework the 0-1 knapsack example discussed in class, but renumbering the objects in the opposite order (so $w_1 = 7, v_1 = 28, \dots, w_5 = 1, v_5 = 1$). All you need to show is a dynamic programming table in the same style as shown in class (slide #10).