

Homework problems:

4.1 (due Sep 27, 2007)

We are given an array A with n elements and a number C . Assume that the sum of the elements in A is larger than C . We would like to compute the smallest number of elements from A whose sum is at least C . (For example, if $A = [8, 3, 9, 2, 7, 1, 5]$ and $C = 18$ then the answer is 3.) Give a linear-time algorithm for this problem. (HINT: use the linear-time SELECT algorithm)

4.2 (due Sep 27, 2007) We are given k sorted lists A_1, \dots, A_k . The total number of elements in the lists is n . We would like to merge the lists into one sorted list B (the number of elements in B will be n). Give an algorithm which solves this problem in time $O(n \log k)$.

4.3 (due Sep 27, 2007) We are given two sorted arrays A and B , each containing n elements. Assume that the arrays do not contain duplicates, and the elements in A are different from elements in B . We would like to compute the median of $A \cup B$. For example if $A = [1, 2, 3, 4, 5]$ and $B = [6, 7, 8, 9, 10]$ then the median is 5; if $A = [1, 3, 5, 7, 9]$ and $B = [2, 4, 6, 8, 10]$ then the median is again 5. Give an $O(\log n)$ algorithm for this problem.

4.4 (due Sep 27, 2007) We are given an array of integers $A[1..n]$. We would like to determine whether there exists an integer x which occurs in A more than $n/2$ times (i. e., whether A has a majority element). Give an algorithm which runs in time $O(n)$. (HINT: use the linear-time SELECT algorithm)

Example: For $A = [3, 1, 2]$ the answer is NO. For $A = [3, 1, 3]$ the answer is YES.

4.5 (due Sep 27, 2007) We are given an array of integers $A[1..n]$. We would like to determine whether there exists an integer x which occurs in A more than $n/3$ times. Give an algorithm which runs in time $O(n)$. (HINT: use the linear-time SELECT algorithm)

4.6 (due Sep 27, 2007) We are given an array of integers $A[1..n]$ which is almost sorted in the following sense: for all $i \in \{1, \dots, n - k\}$ we have $A[i] \leq A[i + k]$. Give an algorithm which sorts the array A . Your algorithm should run in time $O(n \log k)$.

4.7 (due Sep 27, 2007) We are given two arrays of integers $A[1..n]$ and $B[1..n]$, and a number X . Design an algorithm which decides whether there exist $i, j \in \{1, \dots, n\}$ such that $A[i] + B[j] = X$. Your algorithm should run in time $O(n \log n)$.

Preparing for the quiz:

- The next quiz will be on Thursday October 4th. It will cover pages 45-79 of the textbook.
- To prepare for the quiz solve as many problems on pages 70-79 as possible. The following problems are highly recommended: 2.4, 2.5, 2.8, 2.10, 2.12, 2.16, 2.17, 2.18, 2.21, 2.26, 2.27, 2.28, 2.29, 2.30, 2.31, 2.34.
- Please come for the problem session on Fridays 8am, CSB 601 — we will go over the problems (plus any questions you might have).