CS 430 Introduction to Algorithms **Spring Semester, 2021**

Homework 3 version 1.1

Assigned: March 8 *Due: March 22*

Please respect the following guidelines for writing pseudocode:

1. C/Java/Python instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.
2. One instruction per line
3. Match the brackets with a horizontal line
4. Number your lines
5. Write down if your array is indexed $0 \ldots n-1$ or $1 \ldots n$.

Problem 1 Let A and B be arrays of n integers each (do not assume they are sorted). Given an integer x, describe a $O(n \log n)$-time algorithm for determining if there is an integer a in A and an integer b in B such that $x = a + b$.

Present pseudocode and analyze the running time.

Problem 2 Characterize each of the following recurrence equations using the master method (assuming that $T(n) = c$ for $n < d$, for constants $d \geq 1$).

a. $T(n) = 2T(n/2) + (n \log n)^4$

b. $T(n) = 2T(n/2) + \log^2 n$

c. $T(n) = 9T(n/3) + n^2$

d. $T(n) = 9T(n/3) + n^3$

e. $T(n) = 7T(n/2) + n^2$

Problem 3 Show that the running time of QUICKSORT is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.

Problem 4 Describe an O(n)-time algorithm that, given a set S of n distinct numbers and a positive integer $k \leq n$, determines the k numbers in S that are closest to the median of S.

Assume n is odd and the set S is given as an **unsorted** array of size n. You cannot assume the input array is sorted.

Example: if $S = \{1, 3, 5, 9, 13, 21, 101\}$ and $k = 4$, the solution is $\{3, 5, 9, 13\}$. That is, the median itself is included. The answer $\{5, 9, 13, 21\}$ is not correct since 3 is closer to the median (which is 9) than 21. The algorithm should write the output in a separate array, and the numbers **do not** have to be sorted.

You can use the selection algorithm as a subroutine. Precisely, assume that the following procedure is given: SELECT(A,p,q,i) returns (finds) the index j such that $A[j]$ is the i^{th} smallest number among $A[p], A[p+1], \ldots, A[q]$. SELECT correctly runs in time $O(q-p)$ even if the elements of A are not distinct. SELECT here is an extension of Quick-select(A,1,n,i) as in the notes, and only requires two extra tricks for implementation.

Partial credit will be given to correct algorithms, but with larger running time.

Problem 5

1. Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, using the hash function $h(i) = (2i + 5) \bmod 13$ and assuming collisions are handled by chaining. Here the hash table has 13 slots.
2. What is the result of the previous exercise, assuming collisions are handled by linear probing? In the notation from the textbook, use h as defined above for the auxiliary function h'.
3. Show the result of Exercise 1 above in this problem, assuming collisions are handled by quadratic probing, up to the point where the method fails because no empty slot is found. In the notation from the textbook, use h as defined above for the auxiliary function h' and $c_1 = 0, c_2 = 1$. Also, the key will not be inserted if no empty slot is discovered with at most 10 probes.
4. What is the result of Exercise 1 above in this problem, assuming collisions are handled by double hashing using a secondary hash function $h_2(k) = 7 - (k \bmod 7)$? In the notation from the textbook, use $h_1 = h$.