ÇALIŞMA SORULARI 3

1. Given a sorted array of distinct integers A[1, . . . , n], you want to find out whether there is an index i for which A[i] = i.
   a) Write a brute force algorithm. Specify time complexity.
   b) Give a divide-and-conquer algorithm that runs in time O(log n). Prove that the time complexity is in O(logn).

2. a) Write a brute force algorithm to find the maximum difference between any two elements of an n-dimensional array consisting of n numbers. Specify time complexity.
   b) Write a divide and conquer algorithm that finds the maximum difference between any two elements of an n-dimensional array consisting of n numbers in O(n) time. Prove that the time complexity is in O(n).

3. A k-wise merge takes as input k sorted arrays, and constructs a single sorted array containing all of the elements of the input arrays. Describe an efficient divide and conquer algorithm MultiMerge(k, A1, ..., Ak) which computes a k-wise merge of its input arrays. What is the run time of your algorithm with input of k arrays, each of length n.

4. a) Given two integers **x** and **n**, write a function to compute $\mathbf{x^n}$ in O(n) time.
   b)Given two integers **x** and **n**, write a function to compute $\mathbf{x^n}$ in O(logn) time.

5. You are given two sorted lists of size m and n. Give an O(log m + log n) time algorithm for computing the kth smallest element in the union of the two lists.

6. Given a sequence of n elements where each element is an integer in [1,k], return the majority element (an element that appears more than n/2 times) or zero if no majority element is found. Give a divide-and-conquer algorithm that runs in time Θ (nlog n).

7. a) (1D closest pair problem) Given n points in 1-dimension, find two whose mutual distance is smallest. Write a divide and conquer algorithm. Specify time complexity
   b) (2D closest pair problem) Given n points in 2-dimensions, find two whose mutual distance is smallest. Write a divide and conquer algorithm. Specify time complexity

8. Design a greedy algorithm for the assignment problem (see slide about Greedy Algorithms). Does your greedy algorithm always yield an optimal solution? Discuss.

9. Design an algorithm for finding a maximum spanning tree–a spanning tree with the largest possible edge weight–of a weighted connected graph.

10. (Minimum Coin Change Problem) Given a set of coins and a value, find the minimum number of coins which satisfies the value. Write a greedy algorithm. Does your greedy algorithm always yield an optimal solution? Discuss.
For example: coins[] = {5,10,20,25}, value = 50.
Solutions:
{5 * 10} = 50 [10 coins]
{5 * 8 + 10 * 1} = 50 [9 coins] …
Best solution→{25 * 2} = 50 [2 coins]

11. Suppose we are given n ropes of different lengths, and we want to tie these ropes into a single rope. The cost to connect two ropes is equal to sum of their lengths. We want to connect all the ropes with the minimum cost.
For example, suppose we have 4 ropes of lengths 7, 3, 5, and 1. One (not optimal!) solution would be to combine the 7 and 3 rope for a rope of size 10, then combine this new size 10 rope with the size 5 rope for a rope of size 15, then combine the rope of size 15 with the rope of size 1 for a final rope of size 16. The total cost would be 10 + 15 + 16 = 41. (Note: the optimal cost for this problem is 29. How might you combine the ropes for that cost?) Find a greedy algorithm for the minimum cost and prove the correctness of your algorithm.

12. A graph coloring problem is a coloring of the graph vertices s.t. no pair of adjacent vertices share the same color. The chromatic number $\chi(G)$ of a graph G is the smallest number of colors needed to color the graph. Write a greedy algorithm to find the minumum number of colors. Does your greedy algorithm always yield an optimal solution? Discuss.

13. (Coin change problem) Given a set of coins and amount, Write a dynamic programming algorithm to find out how many ways we can make the change of the amount using the coins given. Draw the table using Amount = 5 coins [] = {1,2,3}. Specify time and space complexity.

14. (Rod cutting problem) Suppose you have a rod of length n, and you want to cut up the rod and sell the pieces in a way that maximizes the total amount of money you get. A piece of length i is worth pi dollars. How many ways are there to cut up a rod of length n? (Bruteforce complexity?) Write Write a dynamic programming algorithm that maximizes the total amount of money you get. Draw the table using n=5 and prices={2,5,7,8} for length {1,2,3,4} , respectively.

15. (Edist distance problem) The minimum edit distance between two strings is defined as the minimum number of editing operations (insertion, deletion, substitution) needed to transform one string into another. Write a dynamic programming algorithm to find minimum edit distance between two strings. (The cost of deletion and insertion is +1, substitution is +2, match is 0.)
Draw the table using “intention” and “execution” strings.

16. (Subset sum) Given an array of non-negative integers and an integer sum. We have to tell whether there exists any subset in an array whose sum is equal to the given integer sum. Write a dynamic programming algorithm to solve this subset sum problem. Draw the table using array={3,4,5,2] and sum=6.
Examples:
Input: arr[] = {3, 34, 4, 12, 3, 2}, sum = 7
Output: True
Explanation: There is a subset (4, 3) with sum 7.