**MUH334E18YF** # **/10**

Name: **SOLUTION**

*Each question is worth 5 points.*

**1.** In order to sort the ***n*** songs in your music library, where ***n*** is an even integer, you have come up with an algorithm in which you sort half of your songs and do an additional ***n*** operations.
**a)** What type of an algorithm is this? **Divide-and-Conquer algorithm**
**b)** Find the exact number of operations to sort your library assuming that sorting just one song requires one operation.

**In other words, solve the recurrence relation:** $T(n) = T\left(\frac{n}{2}\right) + n \;\;,\;\; T(1) = 1$ **.**

$$T(n) = T\left(\tfrac{n}{2}\right) + n = T\left(\tfrac{n}{4}\right) + n + \tfrac{n}{2} = T\left(\tfrac{n}{8}\right) + n + \tfrac{n}{2} + \tfrac{n}{4} = \cdots = T(1) + n + \tfrac{n}{2} + \tfrac{n}{4} + \cdots$$
$$= 1 + n\left(1 + \tfrac{1}{2} + \tfrac{1}{4} + \cdots\right) = 2n + 1$$

**2.** ITU gives its students the opportunity to work in part time jobs. Suppose that there are ***m*** jobs offered and ***n*** students applying. Each student is interested in some of the jobs, however one and only one applicant will be hired per job. Your task is to find an assignment of jobs to applicants such that as many applicants as possible get jobs.
Briefly explain your method. *You do not have to explain in detail the algorithm(s) you use. Just state their name(s).*

**This is a Maximum Bipartite Matching problem. To solve it, create a digraph** $G = \{L \cup R \cup \{s,t\} \cup E\}$**, direct all edges (job preferences) from students (L) to jobs (R) and assign unit capacity, add source s and unit capacity edges from s to each node in L, add sink t and unit capacity edges from each node in R to t. Running Ford-Fulkerson's algorithm to find the maximum flow yields the desired assignment.**

**3.** When applying for Master programs in ITU, candidates get to choose 3 programs and rank them as their first, second and last choice. Following the interviews, program coordinators announce their first, second and third accepted candidates. Your task is to design an admission process agreeable for both parties.
**a)** What type of a problem is this? **Stable Matching problem**
**b)** Which algorithm could you use? **Gale-Shapley's Propose-and-Reject algorithm**
**c)** Is the outcome always unique? Explain. **No, the outcome might be applicant-optimal or jury-optimal.**

**4.** You are given the map of Turkey's railway network which shows existing tracks between cities, some of which are one-way only. Your task is to check whether all cities are mutually reachable by train or not.
Briefly explain your method. *You do not have to explain in detail the algorithm(s) you use. Just state their name(s).*

**The task is to check whether the directed graph, where the nodes correspond to cities and the edges correspond to tracks, is strongly connected or not. To do so, one can pick any node, run a BFS (breadth first search), reverse orientation of all edges and run a second BFS. If all nodes are reached in both BFS executions, the graph is strongly connected, i.e. all cities are mutually reachable.**

**5.** During the holidays, you go to your home town to see your relatives who live in ***n*** different houses built in a line. From previous holidays, it is known to you how much pocket money they each will give you. However, this time you cannot visit direct neighbors. Design an algorithm (other than the brute force search) to maximize your allowance and specify its type and time complexity.

**Let i=1 to n represent the relatives (houses) and M[i] the corresponding pocket money. Maximum allowance from the first i houses of the line can be either the maximum allowance from the first i-1 houses of the line or maximum allowance from the first i-2 houses of the line plus the pocket money from the $i^{th}$ house. So, choosing the maximum of these in bottom-up manner yields the desired solution.**

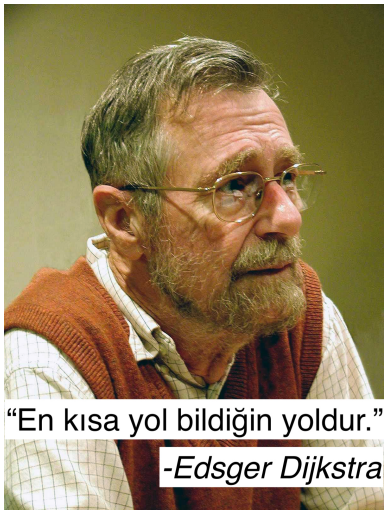
**Pseudocode:**
**Set A[0]=0 and A[1]=M[1].**
**for i=2 to n:**
**A[i] = max { A[i-1] , A[i-2]+M[i] }**

**This dynamic programming solution has a time complexity of O(n).**

**6.** Which unsolved problem in computer science may the excerpt "Are those equal, those who know and those who do not know?" remind of? What is your opinion on this problem?

**It reminds of the P vs NP problem: "Are those equal, those that we know how to solve in polynomial time and those that we do not know how to solve in polynomial time?".**
**Student answers will vary.**

**7.** Which algorithm is referred to in the following meme? What type of an algorithm is this? Briefly explain what it is used for and give an example problem from daily life.




**It refers to Dijkstra's Shortest Path algorithm which is a greedy algorithm to find the simple path of minimum total weight (of its edges) between two given nodes in a directed graph. It can be used for example to find the shortest way to travel from home to university.**

**8.** According to you, what is the most important thing you have learned in this course in regard to your future career?
**Student answers will vary.**