Department of Computer Science and Engineering, SVNIT, Surat

MTech I (1^{st} semester)

Design and Analysis of Algorithms

Lab Assignment No 3 -Greedy Design Technique

AUTUMN Semester 2022-23

Dated Uploaded: 13th Sep 2022

Instructions:

1. *The date of submission **will be specified only two days before**. Therefore, right from the time the assignment is uploaded the students must start implementing the assignments.*
2. *For every delayed submission beyond the deadline, 10 marks per day will be deducted from the maximum marks of the assignment, without any exception, whatsoever may be the scapegoat.*
3. *You can use any programming language, except Python.*
4. *For every program whether specified or not, it is necessary to time your program on input dataset of a large size and give a critique of the theoretical asymptotic analysis of the algorithm that your program is based on and the empirical timing analysis of the program that you have written. The a-priori estimates and the a-posteriori analysis must be in sync with each other.*
5. *All the assignments must be submitted in the form of a zip file containing the Program Source, the screenshot of the output that you obtained, the DataSet/Input Test data used and a ReadMe .txt file explaining what platform to use, what are the input parameters required for execution and how to execute it. Also write your conclusion in ReadMe file. For the theoretical questions in Part B, write the answers using Latex in a .Tex file and submit the .tex as well as .pdf files - to be included in the zip file as above.*
6. *Perform usual error checking. Don't go overboard on this, but don't let your program die because of divide by zero.*
7. *Remember, your programs could be checked by a code-cheating program, so please follow the code of academic integrity.*
8. *There will be a viva for each assignment. This viva would be conducted for this assignment on a future date as specified.*
9. ***Maximum Points: 200***

---

1. Given a string s, remove duplicate letters so that every letter appears once and only once. You must make sure your result is the smallest in lexicographical order among all possible results. Assume that the string **s** consists of small case only and the length of the string is bounded by 10^4.
   - e.g. Input: s = "bcabc", Output: "abc"
   - e.g. Input: s = "cbacdcbc" Output: "acdb"
2. Consider the following variation on the Interval Scheduling Problem. You have a processor that can operate 24 hours a day, every day. People submit requests to run daily jobs on the processor. Each such job comes with a start time and an end time; if the job is accepted to run on the processor, it must run continuously, every day, for the period between its start and end times. (Note that certain jobs can begin before midnight and end after midnight; this makes for a type of situation different from what we saw in the Interval Scheduling Problem). Given a list of n such jobs, your goal is to accept as many jobs as possible (regardless of their length), subject to the constraint that the processor can run at most one job at any given point in time. Provide an algorithm to do this with a running time that is polynomial in n. You may assume for simplicity that no two jobs have the same start or end times. Example. Consider the following four jobs, specified by (start-time, end-time) pairs. $(6P.M., 6A.M.), (9P.M., 4A.M.), (3A.M., 2P.M.), (1P.M., 7P.M.)$ The optimal solution would be to pick the two jobs $(9P.M., 4A.M.)$ and $(1P.M., 7P.M.)$, which can be scheduled without overlapping.

3. Suppose the job of a firm is to manage the construction of billboards on the Jammu-Srinagar Highway runs South-North for M kms. The possible sites for the billboards are given by numbers $x_1, x_2, x_3, x_4, x_5, \ldots . \ x_n$ each in the interval $[0 \ldots M]$. x_i's are indicating the position of the billboards along the Highway in kms, measured from its western end. If it is decided to place a billboard at location x_i, then the firm receives a revenue of $r_i > 0$. Regulations of the JK Govt require that no two of the billboards be within less than or equal to 5 kms of each other. Thus, as part of the optimization problem, the firm has to decide where to place the billboards at a subset of the sites $x_1, x_2, x_3, x_4, x_5, \ldots . \ x_n$ so as to maximize the total revenue, subject to this constraint. Give an algorithm that takes in an instance of this problem as input and returns the maximum total revenue that can be obtained from any valid subset of sites. Give its running time also. Implement the algorithm that you designed and analyze its time complexity.

   An example : Suppose $M = 20$, n = 4 , $[x_1, x_2, x_3, x_4] = [6, 7, 12, 14]$, $[r_1, r_2, r_3, r4] = [5, 6, 5, 1]$. Then, what would be the optimal solution ?

4. Suppose that we have a set of activities to schedule amongst a large number of lecture halls. We wish to schedule all the activities using as few lecture halls as possible. Give an efficient greedy algorithm to determine which activity should use which lecture hall and implement this algorithm.

5. This concerns the first activity selection problem discussed in the class with the objective function to select maximal set of mutually compatible activities out of a given set of activities with a defined start time and finish times. Suppose that instead of selecting the first activity to finish, we instead select the last activity to start that is compatible to all the previous activities. Write a program that implements this algorithm and analyze its time complexity. Also, write a program that implements the optimal algorithm discussed in the class and analyze the time complexity. Compare the time complexities of the two different programs that you have implemented.

6. Consider the 0/1 knapsack problem with n objects whose profits and weights are v_i, w_i 1≤i≤n, respectively. The capacity of the knapsack is m. IT is also given that all the objects have the same weight. Present an $O(m)$ algorithm to solve this problem and implement it.

7. Suppose that in a 0-1 knapsack problem, the order of the items when sorted by increasing weight is the same as their order when sorted by decreasing value. Give an efficient algorithm to find an optimal solution to this variant of the knapsack problem and and implement it.

8. Let F(I) be the value of the solution generated on knapsack problem instance I by GreedyKnapsack when the objects are processed in nonincreasing order of their profit values. Let $F*(I)$ be the value of an optimal solution of this instance. How large can the ratio $F*(I)/F(I)$ get. Show with a parameterized implementation of this problem wherein apart from other data, the ratio $F*(I)/F(I)$ is given as input in terms of a positive integer.

9. The Director, IIT Jammu drives an automobile from Jammu to New Delhi along the Jammu Delhi highway. His car's petrol tank, when full, holds enough gas to travel n kms and his map gives the distances between petrol pump stations on his route. The Professor wishes to make as few petrol station stops as possible along the way. Give an efficient method by which Professor can determine at which petrol stations should he stop and implement the same.

10. Let's consider a long, quiet country road with houses scattered very sparsely along it. That is, picture the road as a line segment, with an eastern endpoint and a western endpoint). Further, let's suppose that despite the bucolic setting, the residents of all these houses are avid cell phone users. You want to place cell phone base stations ('towers ' as we say here) at certain points along the road, so that every house is within four miles of one of the base stations.

    Give an efficient algorithm that achieves the goal,using as few base stations as is possible. Implement your algorithm.

11. Suppose a Consulting firm that does security consultancy needs to obtain licenses for n different pieces of cryptographic software. However, due to regulations, they can only buy these licenses at the rate of at most one license per month. Each license is currently selling at the rate of Rs 1000. However, the cost of the licenses are designed to increase following the monthly exponential growth, at the defined rates i.e. the cost of license j increases at the rate r_j (> 1) every month, that grows exponentially. For example, if license j is purchased t months from now, it will cost 100 * r_{jt}.We assume that all the price growth rates are distinct i.e. r_j != r_j for i != j. Given as the input, the n rates of price growth viz. r_1, r_2, r_3, ...r_n,

determine the order in which the licenses must be bought so that the total amount of money spent is minimized. Implement in the form of a program.

12. An Euler circuit in a directed graph is a cycle in which every edge is visited exactly once. Design a linear-time algorithm to find an Euler circuit in a directed graph where one exists, and implement the same.

13. You are given a set of N sticks, which are lying on top of each other in some configuration. Each stick is specified by its two endpoints; each endpoint is an ordered triple giving its x, y, and z coordinates; no stick is vertical. A stick may be picked up only if there is no stick on top of it.
    (a) Explain how to write a routine that takes two sticks a and b and reports whether a is above, below, or unrelated to b.
    (b) Give an algorithm that determines whether it is possible to pick up all the sticks, and if so, provides a sequence of stick pickups that accomplishes this.

14. A graph is k-colorable if each vertex can be given one of k colors, and no edge connects identically colored vertices. Give a linear-time algorithm to test a graph for two-colorability. Assume graphs are stored in adjacency-list format; you must specify any additional data structures that are needed. How is a two-colorable graph related to a bipartite graph ? Give at least two applications of the two-colorability or that of bipartite graph - the distinctiveness of your application could fetch more marks.

15. Write a program to implement file compression (and uncompression) using Huffman's algorithm.

16. Edge-weighted graphs can be interpreted as a network of pipes, where the weight of edge (i, j) determines the capacity of the pipe. Capacities can be thought of as a function of the cross-sectional area of the pipe. A wide pipe might be able to carry 10 units of flow in a given time, where as a narrower pipe might only carry 5 units. The network flow problem asks for the maximum amount of flow which can be sent from vertices s to t in a given weighted graph G while respecting the maximum capacities of each pipe.

    The solution to the network flow problem can be used to solve the following problem: viz. bipartite matching. A matching in a graph G = (V, E) is a subset of edges E' $\subseteq$ E such that no two edges of E' share a vertex. A matching pairs off certain vertices such that every vertex is in, at most, one such pair, as shown in Figs 1 and 2.

    The algorithm that you will have to implement is as follows:

    - Create a source node s that is connected to every vertex in L by an edge of weight 1.
    - Create a sink node t and connect it to every vertex in R by an edge of weight 1.
    - Finally, assign each edge in the bipartite graph G a weight of 1.
    - Now, the maximum possible flow from s to t defines the largest matching in G. Certainly one can find a flow as large as the matching by using only the matching edges and their source-to-sink connections. Further, there can be no greater possible flow.

17. A person wants to fly from city A to city B in the shortest possible time. He turns to the travelling agent who knows arrival and departure times of all the flights in the world. Design an algorithm that will allow the agent to choose a route with the minimum total travel time.

18. There are n people, each in possession of a different rumor. They want to share all the rumors with each other by sending electronic messages. Assume that a sender includes all the rumors he or she knows at the time the message is sent and that a message may only have one addressee. Design a greedy algorithm that always yields the minimum number of messages they need to send to guarantee that every one of them gets all the rumors.

19. Design and conduct an experiment to empirically compare the efficiencies of Prim's and Kruskal's algorithms on random graphs of different sizes and densities.

20. Four villages are located at the vertices of a unit square in the Euclidean plane. You are asked to connect them by the shortest network of roads so that there is a path between every pair of the villages along those roads. Find such a network. Design and implement an algorithm that computes such a network.

*****Ends*****

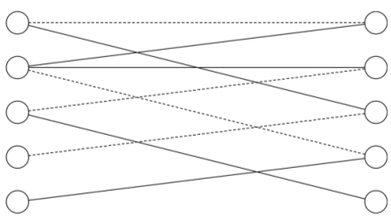

Figure 1: Bipartite Matching

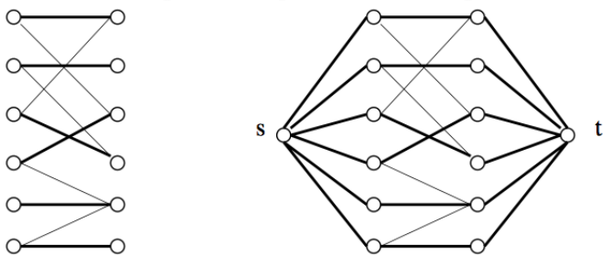

Figure 2: Bipartite graph with a maximum matching highlighted (on left). The corresponding network flow instance highlighting the maximum s - t flow (on right).