CS21003 ALGORITHMS-1
(Tutorial 6 : Greedy Strategy)
Date: October 17, 2020

Question 1:

You are given a sequence of n songs where the i^{th} song is L_i minutes long. You want to place all of the songs on an ordered series of CDs (e.g. $CD_1, CD_2, CD_3, ..., CD_k$) where each CD can hold m minutes. Furthermore the songs must be recorded in the given order, song 1, song 2, ..., song n. All songs must be included. No song may be split across CDs. Your goal is to determine how to place them on the CDs as to minimize the number of CDs needed.

- ▷ Give the most efficient algorithm for this problem.
- ▷ Prove the correctness of the algorithm and analyze the time complexity.

Question 2:

You are the manager of Grand Budapest Hotel. You are given the arrival and departure time of each guest. But your cleaning staff is lazy. So you have to instruct them to clean minimum number of rooms. If there are two guests and their stay in the hotel does not coincide then you need only one cleaned room because after 1st guest leaves you can give the same room to 2nd guest. So for given arrival and departure time of each guest design optimal algorithm to return minimum number of rooms to be cleaned by the cleaning staff. Analyze the time complexity of the algorithm. You may assume departure time is strictly greater than arrival time and all time instances provided are distinct.

Question 3:

The queen of Dragons has waged a war against the army of the dead (the wights). The wights are marching in a straight line and approaching the wall. The queen plans to send her three dragons(Drogon, Rhaegal and Viserion) to defeat the army of the dead. The cost of defeating each wight is different for each dragon and No two consecutive wights should be defeated by the same dragon. As the chief strategist, queen has asked you to come up with an optimal strategy that would reduce the total cost.

Question 4:

Let's consider a long, quiet country road with houses scattered very sparsely along it. (We can picture the road as a long 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 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 this goal, using as few base stations as possible.

Question 5:

A small business—say, a photocopying service with a single large machine—faces the following scheduling problem. Each morning they get a set of jobs from customers. They want to do the jobs on their single machine in an order that keeps their customers happiest. Customer i's job will take t_i time to complete. Given a schedule (i.e., an ordering of the jobs), let C_i denotes the finishing time of job i. For example, if job j is the first to be done, we would have $C_j = t_j$; and if job j is done right after job i, we would have $C_j = C_i + t_j$. Each customer i also has a given weight w_i that represents his or her importance to the business. The happiness of customer i is expected to be dependent on the finishing time of i's job. So the company decides that they want to order the jobs to minimize the weighted sum of the completion times, $\sum_{i=1}^{n} w_i C_i$

Design an efficient algorithm to solve this problem. That is, you are given a set of n jobs with a processing time t_i and a weight w_i for each job. You want to order the jobs so as to minimize the weighted sum of the completion time.

Question 6:

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: Given the following jobs- (6 P.M. , 6 A.M.), (9 P.M. , 4 A.M.), (3 A.M. , 2 P.M.), (1 P.M. , 7 P.M.) The optimal solution would be to pick the two jobs (9 P.M. , 4 A.M.) and (1 P.M., 7 P.M.), which can be scheduled without overlapping.

Question 7:

You are consulting for a trucking company that does a large amount of business shipping packages between New York and Boston. The volume is high enough that they have to send a number of trucks each day between the two locations. Trucks have a fixed limit W on the maximum amount of weight they are allowed to carry. Boxes arrive at the New York station one by one, and each package i has a weight w_i. The trucking station is quite small, so at most one truck can be at the station at any time. Company policy requires that boxes are shipped in the order they arrive. At the moment, the company is using a simple greedy algorithm for packing: they pack boxes in the order they arrive, and whenever the next box does not fit, they send the truck on its way. But they wonder if they might be using too many trucks, and they want your opinion on whether the situation can be improved. Here is how they are thinking. Maybe one could decrease the number of trucks needed by sometimes sending off a truck that was less full, and in this way allow the next few trucks to be better packed. Prove that, for a given set of boxes with specified weights, the greedy algorithm currently in use actually minimizes the number of trucks that are needed.