CS 330 – Fall 2019 – Lab 4

Question 1. ***Non-tree edges of BFS***
Consider running BFS on an undirected graph. Edges that do not appear in the BFS trees have special properties.

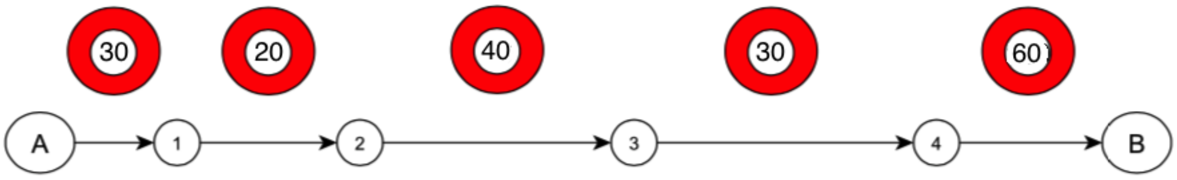
i) *Where are non-tree edges? Given the BFS tree, how do you find non-tree edges?*

ii) *Can you identify the non-tree edges on-the-fly? That is, can you augment the BFS algorithm to find them?*

iii) *Can you generalize the result to directed graphs?*

Question 2. ***Job Overlapping***
The manager of a large student union on campus comes to you with the following problem. She's in charge of a group of n students, each of whom is scheduled to work one shift during the week. There are different jobs associated with these shifts (tending the main desk, helping with package delivery, rebooting cranky information kiosks, etc.), but we can view each shift as a single contiguous interval of time. There can be multiple shifts going on at once. She's trying to choose a subset of these n students to form a super- vising committee that she can meet with once a week. She considers such a committee to be complete if, for every student not on the committee, that student's shift overlaps (at least partially) the shift of some student who is on the committee. In this way, each student's performance can be observed by at least one person who's serving on the committee. Give an efficient algorithm that takes the schedule of n shifts and produces a complete supervising committee containing as few students as possible.

Question 3. ***Biker in a hurry!***



Suppose that a biker wants to go from a city A to a city B, and drives over a straight highway that is split in n segments. Each segment i has a length l_i and a speed limit u_i. The biker -because there is an urgent need- has a special permission, that allows her to violate the speed limit for a total time T, and for a speed v. She has to choose therefore, in which segments $i_1, i_2, \ldots$ of her path and for how long in terms of time, she will travel with speed $u_{i_1} + v, u_{i_2} + v, \ldots$, such that the total time that she violates the speed limits is T, and the duration of her trip from city A to city B is minimized.

i) *Argue that the time needed to traverse a segment i, if she violates the speed limit for u mph and for a time t_i, is:*

$$dt_i = \frac{l_i}{u_i} - \frac{v}{u_i} \cdot t_i$$

ii) *Devise a greedy algorithm that will allow the biker to go from city A to city B in the minimum time, if she is allowed to violate the speed limit for a total time T and for v mph. You don't need to show the correctness for now.*
Hint: Observe from (i), which quantity affects the time she needs to travel over a segment

Question 4. ***Job scheduling***
IS&T services at BU has some shared resources available for students and faculty. To make use of these resources each person has to submit the exact time that she intends to run her job and how long it will take. Then IS&T will decide to allow some of the jobs to run but will reject others. Their goal is to fulfill ***as many requests as possible*** *with the caveat that* ***only one job at a time*** *can run. (Jobs can only be started at the exact time that was requested, otherwise they are rejected.)*

In the following, you will devise an algorithm to decide which jobs to run or reject to maximize the number of jobs that are accepted. We say that a schedule is valid *if it has no overlapping jobs. The schedule is* optimal *if it has the maximum number of jobs.*

i) *Assume the requests are given as the tuples (start time, duration in mins) below. Which jobs should be included in an optimal schedule? Note, that there is more than one optimal schedule.*

- *job_1 (9:00, 20 m)*
- *job_2 (9:15, 10 m)*
- *job_3 (8:00, 65 m)*
- *job_4 (10:00, 120 m)*
- *job_5 (9:30, 40 m)*
- *job_6 (10:15, 60 m)*
- *job_7 (9:55, 70 m)*
- *job_8 (10:00, 12 m)*
- *job_9 (10:00, 10 m)*

ii) *Compare the two optimal solutions that you found. Find a pair of jobs in the two schedules, so that you can exchange one for the other and still have a valid schedule. Why were you able to make the exchange? Try to formulate why it is always true, that if you have more than one optimal schedule, then you can always find such a pair of jobs to switch between two optimal schedules.*