CS 331: Algorithms and Complexity (Spring 2015)

Unique numbers: 51825, 51830, 51835, 51840

Assignment 3

Due on: Wednesday, April 1^{st}, by 11.59pm

Q1 Decide whether each of the following claims is true or false (please state the conclusion for each claim).

If it is true, give a rigorous proof. If it is false, give a explicit counterexample (it is allowed to assume the order of edges in the execution of any specific algorithm).

[a] (5 points) Consider an implementation of the Ford-Fulkerson algorithm which does not create any backward edges in the residual graph. It is claimed that, there exists a constant $\alpha > 0$, such that for any flow network G, this modified implementation is guaranteed to find a flow of value at least α times the maximum-flow value in G.

Let $G = (V, E)$ be a network with a source s, a sink t and a capacity c_e on every edge $e \in E$. Bob has two claims:

[b] (5 points) Consider a network $G = (V, E)$ with a source s, a sink t and a capacity c_e on every edge $e \in E$. The claim is that, if f is a maximum $s - t$ flow in G, then f either saturates every edge out of s with flow, or saturates every edge into t with flow (f saturates an edge means $f(e) = c_e$).

[c] (5 points) Consider a network $G = (V, E)$ with a source s, a sink t and a capacity c_e on every edge $e \in E$. Let A, B be a minimum $s - t$ cut with respect to the capacities c_e. The claim is that, If we add 1 to every capacity, namely $c'_e = c_e + 1$ for every $e \in E$, then A, B is still a minimum $s - t$ cut with respect to the new capacity c'_e.

Q2 Alice and Bob just opened a transportation company. They have n drivers and n buses. They know every driver i has a list $L_i \subset \{1, \cdots, n\}$ of buses that he could operate. They already have a perfect plan inwhich every driver i is assigned to some bus $b_i \in L_i$. Elements of $\{b_1, \cdots, b_n\}$ are distinct, and no bus is vacant.

Now, they want to add a new bus $n + 1$ next week and hire a new driver. They have updated the lists L_i for all drivers, including the new driver. But the problem is L_{n+1} does not contain bus $n + 1$, and hence the new driver cannot be assigned to the new bus. Some assignments need to be changed.

[a] (8 points) Given $n, \{b_1, \cdots, b_n\}$ and $L_1, \cdots, L_n, L_{n+1}$ such that $\{b_1, \cdots, b_n\}$ are distinct elements from 1 to n and every $L_i \subset \{1, \cdots, n + 1\}$. Alice and Bob are looking for a schedule such that these $n + 1$ drivers are enough to operate every bus and each of them only drive a bus in his list. At the same time, they also hope the modification on the current schedule is as small as possible. Please give a efficient algorithm to determine whether there exists a new schedule, if so, please find out any one with the **smallest modification**.

Example 1: Consider the input: $n = 2, b_1 = 1, b_2 = 2, L_1 = \{1, 3\}, L_2 = \{1, 2\}, L_3 = \{1, 2\}$.

There exists two new schedules $\{(b_1, b_2, b_3) = (3, 1, 2), (3, 2, 1)\}$. The schedule $b_1 = 3, b_2 = 2, b_3 = 1$ has the smallest modification.

Example 2: Consider the input: $n = 3, b_1 = 1, b_2 = 3, b_3 = 2, L_1 = \{1, 4\}, L_2 = \{2, 3\}, L_3 = \{2\}, L_4 = \{3\}$.

There does not exist a new schedule, because L_2, L_3, L_4 only contains 2 options.

[b] (7 points) Alice and Bob have also decided to buy some GPSes for their company such that they can track buses (currently they do not own any GPSes). They could either install a GPS in one bus, or assign a GPS to one driver such that the driver will carry it in working time for tracking.

Right now, they do not have any GPS on the buses and no driver owns a GPS. After installing a GPS on a bus or assigning a GPS to one driver, they cannot remove it or take it away. They want to achieve this point: for any driver i and any bus j in his list L_i, either driver i carries a GPS or there is a GPS installed in bus j such that they could track it no matter what happens. Given n and $L_1, \cdots, L_n, L_{n+1}$ that every $L_i \subset \{1, \ldots, n+1\}$, please provide an efficient algorithm to find the smallest number of GPSes they need and provide a scheme that satisfies the requirements mentioned above with the smallest number of GPSes.

Example 1: Consider the input: $n = 2, L_1 = \{1, 3\}, L_2 = \{1, 2\}, L_3 = \{1, 2\}$.

They need 3 GPSes. Either every driver brings a GPS or they installed a GPS on every bus.

Example 2: Consider the input: $n = 3, L_1 = \{1, 4\}, L_2 = \{2, 3\}, L_3 = \{2\}, L_4 = \{3\}$.

Using 4 GPSes is easy: either every driver brings a GPS or they installed a GPS on every bus. However, at most 3 buses will be operated simultaneously, it is possible to use 3 GPSes. Actually the optimal solution needs 3 GPSes on driver 1, bus 2 and bus 3. Installing GPSes on bus 1,2,3 is not a valid solution because the first driver may operate bus 4.

Hint 1: The number of GPSes needed depends on the first answer.

Hint 2: To find out the scheme, let P be the drivers to bring a GPS and Q be the buses installed a GPS. For convenience, let $\bar{P}$ and $\bar{Q}$ denote the complements of P and Q. From the requirement in the problem, there is no edge between $\bar{P}$ and $\bar{Q}$. So try to construct a bipartite graph such that $\{\bar{P}, Q\}$ and $\{P, \bar{Q}\}$ constitute a cut of the bipartite graph.

For both part (a) and part (b), show the running time of the algorithms and provide a proof of correctness. It is recommended to provide a short piece of pseudo-codes (like 5-10 lines) for every algorithm.