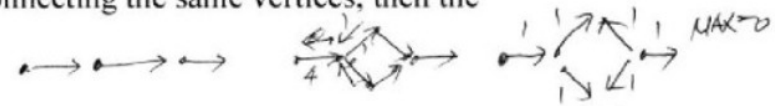
Q1 18

1) 20 pts

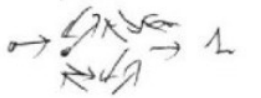
Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[TRUE/FALSE]
In a flow network, a flow f is a max flow if and only if there exists no cut with the same capacity as $v(f)$

[TRUE/FALSE]
If we replace each directed edge in a flow network with two directed edges in opposite directions with the same capacity and connecting the same vertices, then the value of the maximum flow remains unchanged.



[TRUE/FALSE]
It is possible for a circulation network to not have a feasible circulation even if all edges have unlimited capacity.



[TRUE/FALSE]
If subset sum can be solved in polynomial time, then 3SAT can be solved in polynomial time.

[TRUE/FALSE]
If problem A is NP-complete and $A \leq_p B$, then problem B is also NP-complete.

[TRUE/FALSE]
Some problems in the set P do not have efficient certifiers.

[TRUE/FALSE]
If an NP-hard problem can be solved in polynomial time then P=NP

[TRUE/FALSE]
A dynamic programming algorithm always uses some type of recurrence relation

[TRUE/FALSE]
The main difference between divide and conquer and dynamic programming is that divide and conquer solves problems in a top-down manner whereas dynamic-programming does this bottom-up.

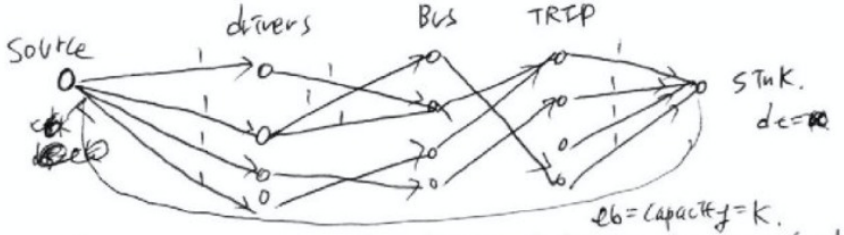
[TRUE/FALSE]
A dynamic programming solution will always have a polynomial running time.

Q5 **14**

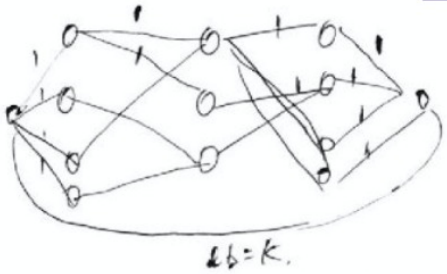
5) 15 pts

A bus rental company needs to solve the following problem daily and they need an efficient algorithm to help with their planning. They have n buses, m drivers, and they need to plan for k trips. Each bus is suitable only for certain types of trips. For example, a trip to the mountains may require shorter and more powerful buses. They also know that not every driver can drive every bus. Note: a driver and a bus are assigned to a trip for the whole day. n>k, m>k

a- Describe a Network Flow based solution to determine, given k trips (with type of trip specified) if they can satisfy all k trips for that day. (10 pts)



okay **10**



① Connect source to drivers / drivers to Bus / and Bus to trips.
(there is a relation between drivers & bus → Bus and TRIP)
Set 1 to all capacities in edges.

② Connect sink to source and set value (lower bound =k, capacity =k)

③ we can check ~~flow~~ is k. or not.
the circulation since there must be a flow (k)
due to from sink to source (k)
edge

b- If they cannot meet all the demand for trips on a given day, they want to know whether it is due to the fact that they don't have an appropriate set of drivers or a suitable set of buses, or both. Describe how they can make that determination. (5 pts)

we can check ~~the compare table~~ min cut (driver, bus) and cut (bus to TRIP)

① we can check the total value of flow between driver nodes and bus nodes using cut between them.

if the v(f) < k, it means there is not enough demand for drivers.

missing a case for "both" **4**

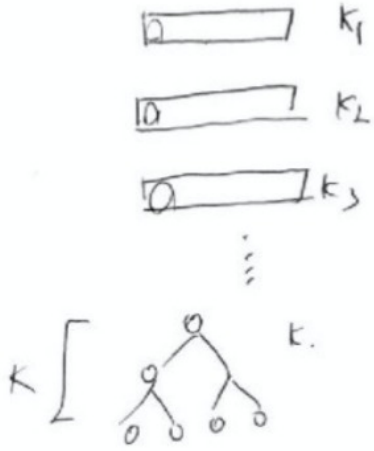
② we can check the flow between Bus to TRIP using cut between them.

If v(f) between drivers and bus ≥ k but v(f) between bus and trip < k, Then it means there is not enough bus to trip.

Q6 10

6) 10 pts

In the MERGE-SORT algorithm we merge two sorted lists into one sorted list in O(n) time. Describe an O(n log k)-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. Be sure to explain why your algorithm runs in O(n log k)-time.



we can create k size of MIN Heap
and put ~~this~~ first value from Each List. ($K_1(1), K_2(1), K_3(1)\ldots$)

The total number of elements is n, which means that we add value to min heap n times, and sort this heap log k times (since the size of heap is k).
(Sort to min heap takes log k times)
This takes n * log k time to combine these all Sorted list ~~and~~ to one sorted list.

$O(n \log k)$ + → n log k (sort, cluster min heap, extract value)
+ n (assign value to array)
size of n.

10

∴ $O(n \log k)$