Due: Tuesday, April 11, 2023, 11:59pm

Submit online at (clickable link): `https://forms.gle/xJWWzL7cDGmcxzHG7`

1 Shipping Coffee

You work for a company that has extensive coffee operations alongside a river. Each day, coffee beans harvested in plantations upriver must be brought to processing plants downriver. The coffee beans are hand-picked, packed in standard containers and brought to the nearest port on the river. Your company has contracts with individual barge captains to pick up the containers and bring them downriver to the ports near the processing plants.

Unfortunately, each captain has docking privileges only at certain ports. It is your job to coordinate which captain will pick up coffee at which ports and where the coffee will be delivered. Furthermore, you have to contend with the following parameters and constraints:

- Your company has n coffee plantations, each one is near a different upriver port.
- Your company has m processing plants, each one is near a different downriver port.
- Your company has a fleet of k barges.
- The barges are slow and can only make one trip from upriver to downriver each day.
- Each barge can dock at multiple upriver ports and pick up coffee containers from multiple plantations and then float down river and drop off the coffee at multiple processing plants. However, there is not enough time for the barge to travel back upriver to pick up more coffee during the remainder of the day.
- Each plantation produces 10 containers of coffee each day.
- Each processing plant can process 75 containers of coffee each day.
- Each barge can carry 25 containers.
- For each barge captain, you have a list of upriver ports and a list of downriver ports where the captain may dock.

Assignment: Use max flow to produce a solution that maximizes the total number of containers of coffee brought from the plantations to the processing plants without exceeding the capacity of the barges and without delivering more coffee to a processing plant than it can process. Do the following:

1. Using a combination of a diagram, descriptive labels and English sentences, describe how you can transform the problem described above into a flow network.
2. Argue that your transformation works. That is, the maximum flow of the flow network should tell you which ports each barge must stop at to pick up coffee (and how much to pick up) and which ports each barge must stop at to deliver coffee (and how much to deliver).
3. Explain why the solution your algorithm produced is the best possible solution.

2 Core is NP-Complete

Let $G = (V, E)$ be an undirected graph. We say that $C \subseteq V$ is a *core* of G if for every vertex $u \in V - C$ there exists $v \in C$ such that $(u, v) \in E$. That is, every vertex in G is either already in the core, C, or is adjacent to some vertex in C. In this problem, you will show that it is NP-complete to determine whether a graph has a small core. Formally, we define Core by:

Core

INPUT: an undirected graph $G = (V, E)$ and natural number k.

QUESTION: Does G have a core C, where $|C| \leq k$? I.e, does there exist $C \subseteq V$ such that $|C| \leq k$ and for every $u \in V - C$ there exists $v \in C$ such that $(u, v) \in E$?

Recall that Vertex Cover was shown to be NP-complete in lecture:

VC

INPUT: an undirected graph $G = (V, E)$ and natural number k.

QUESTION: Does G have a vertex Cover V', where $|V'| \leq k$? I.e, does there exist $V' \subseteq V$ such that $|V'| \leq k$ and for every edge $(u, v) \in E$, we have $u \in V'$ or $v \in V'$?

Show that Core is NP-complete by constructing a $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduction f from VC to Core.

1. Give an example of an undirected graph G and a number k such that $(G, k) \in$ Core but $(G, k) \notin$ VC. You can thus conclude that the identity function does not reduce VC to Core.
2. Describe a polynomial-time function f that given input (G_1, k_1) outputs (G_2, k_2) where G_1 and G_2 are undirected graphs and k_1 and k_2 are natural numbers.

   N.B.: the reduction function f has (G_1, k_1) as its **ONLY** input. It has no other information. In particular, it does not know whether G_1 has a vertex cover with $\leq k_1$ vertices.
3. Briefly argue that f runs in time polynomial in the size of G.
4. Prove that if $(G_1, k_1) \in$ VC then $(G_2, k_2) \in$ Core where $(G_2, k_2) = f((G_1, k_1))$ for the function f you described in part 1.
5. Prove that if $(G_2, k_2) \in$ Core then $(G_1, k_1) \in$ VC where $(G_2, k_2) = f((G_1, k_1))$ for the function f you described in part 1.

N.B.: the proofs in parts 4 and 5 must be separate proofs. Also, these proofs must hold for all undirected graphs G, not just conveniently chosen examples. (That's why a proof is required.)

3 Hamiltonian Cycle vs Hamiltonian Path

The Hamiltonian Cycle (HC) and the Hamiltonian Path (HP) problems are similar, but different:

Hamiltonian Cycle (HC)

Input: an undirected graph $G = (V, E)$

Decide: Is there a cycle in G that visits every vertex in V exactly once?

Hamiltonian Path (HP)

Input: an undirected graph $G = (V, E)$ and two vertices $s \in V$ and $t \in V$, where $s \neq t$.

Decide: Is there a path from s to t in G that visits every vertex in V exactly once?

Construct a $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduction f from HC to HP. Then, prove your function f is indeed a polynomial-time many-one reduction. Use the following steps:

1. Give a high-level description of a polynomial-time function f that given an undirected graph G, outputs a graph G' and two vertices s and t.

   N.B.: the reduction function f has G as its **ONLY** input. It has no other information. In particular, it does not know whether G has a Hamiltonian Cycle. Furthermore, f must output G', s and t in the case where G has a Hamiltonian Cycle and when G does not have a Hamiltonian Cycle. Specifically, do not start your description with "Let (u, v) be an edge in a Hamiltonian Cycle of G."

2. Briefly argue that f runs in time polynomial in the size of G.

3. Prove that for all undirected graphs G and $(G', s, t) = f(G)$ if G has a Hamiltonian Cycle, then there is a Hamiltonian Path from s to t in G'.

4. Prove that for all undirected graphs G and $(G', s, t) = f(G)$ if there is a Hamiltonian Path from s to t in G' then G has a Hamiltonian Cycle.

N.B.: the proofs in parts 3 and 4 must be separate proofs. Also, these proofs must hold for all undirected graphs G, not just conveniently chosen examples. (That's why a proof is required.)