Midterm 1

Name:

Penn State ID:

Student ID number:

Instructions: Answer all questions (no % 20 option on the exam). Read them carefully first. Be precise and concise. Handwriting needs to be neat (if we can't understand what you write, we can't grade it). Box numerical final answers. Write only on the provided space; if you need extra space for a question, put a note in the question and use the extra pages provided at the end.

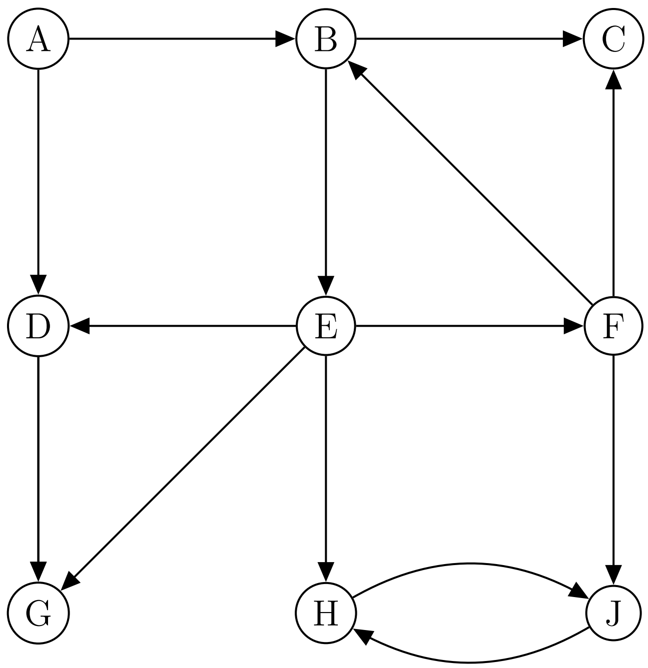
Please clearly write your name and your PSU Access User ID (i.e., xyz1234) in the box on top of every page.

Good luck!

Name:	PSU Access ID:

1 Strongly Connected Components *(20 points)*

(a) Give the strongly connected components of the graph below in the order in which they would be discovered by the algorithm discussed in class (DFS in the reversed graph, then DFS on the graph, processing nodes, and edges out of a node, in alphabetic order.). Give the connected components in the form ABC, DEFG, etc.,



Solution

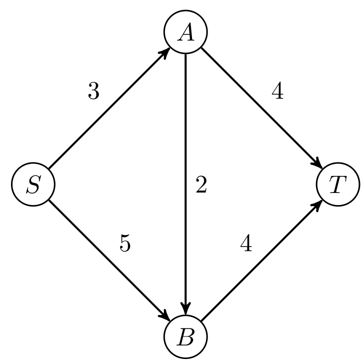
We first run DFS on the reverse graph to sort the nodes in descending order by post-number. This gives the following order to the nodes: H, J, G, D, C, B, F, E, A.

We then run DFS on the regular graph using this order to obtain the following SCCs: HJ, G, D, C, BEF, A.

Name:	**PSU Access ID:**

2 Maximum flow *(20 points)*

(a) Find the maximum flow and the minimum cut in the following flow network (the source node S and the sink node is T).

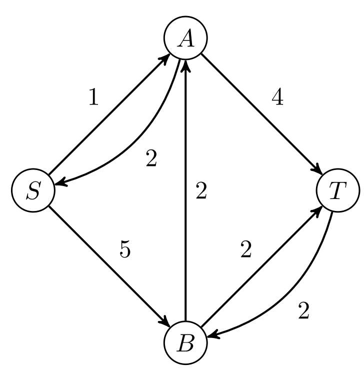


Solution

Max Flow is 7. Min Cut is {S, B}, {A, T} (cutting the edges (S,A) and (B,T)).

(b) Suppose that in the first augmentation of the Ford-Fulkerson algorithm you choose the augmenting path $S \to A \to B \to T$. What is the residual graph after this? (draw it)

Solution



Name:	PSU Access ID:

(c) Complete this sentence, and justify your answer briefly:

In a flow network, increasing the capacity of an edge (A, B) by 1 results in an increase in the value of the maximum flow if and only if...

Solution: "The edge (A, B) belongs to every minimum cut of the flow network".

If there is a unique minimum cut containing (A, B) then increasing the capacity of (A, B) increases the capacity of the cut and the value of the maximum flow. Conversely, if there is a minimum cut that does not contain (A, B), then the capacity of the cut will remain unchanged after the capacity of (A, B) increases. If the capacity of the min cut does not changes, neither does the maximum flow.

Alternate Correct Solution: there exists an augmenting path in the residual graph resulting from pushing a maximum flow of the original flow network through the flow network modified so that the capacity of (A, B) is increased by 1.

Name:	PSU Access ID:

3 Linear Programming: the basics *(20 points)*

$$\begin{aligned} \max \quad & x_1 + x_2 + x_3 \\ \text{subject to:} \quad & x_1 + x_2 \le 4 \\ & x_1 + x_3 \le 2 \\ & x_2 + 2x_3 \le 1 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

(a) Write the dual linear program (LP) of the LP above; use variables $y_1, y_2 \dots$ etc.

Solution

$$\begin{aligned} \min y_1 + y_2 {+} y_3 \\ \text{subject to } y_1 + y_2 \ge 1 \\ y_2 + y_3 \ge 1 \\ y_1 + y_3 \ge 1 \\ y_1, y_2, y_3 \ge 0 \end{aligned}$$

(b) Simplex (for the original LP) starts at the point $(0, 0, 0)$. In the first step, it increases the variable x_2. What happens next? Rewrite the LP in the new coordinate system.

Solution Our new variable values are $(0, 1, 0)$ and we are now tight on the constraints $x_1 \ge 0$, $x_2 + 2x_3 \le 1$, and $x_3 \ge 0$ and loose on $x_2 \ge 0$. Our new coordinate system has the axes $z_1 = x_1 = 0$, $z_2 = 1 - x_2 - 2x_3$, $z_3 = x_3 = 0$.

$x_2 = 1 - z_2 - 2z_3$

$$\max z_1 - z_2 - z_3 + 1$$

$$\begin{aligned} z_1 - z_2 - 2z_3 &\le 3 \\ z_1 + z_3 &\le 2 \\ z_2 + 2z_3 &\le 1 \\ z_1, z_2, z_3 &\ge 0 \end{aligned}$$

Name:	PSU Access ID:

4 Linear Programming: short questions *(20 points)*

(a) There are several major cities and water reservoirs in Pennsylvania.

- Reservoir i holds G_i gallons of water, while city j needs D_j gallons of water.
- It costs p_{ij} dollars per gallon of water supplied from reservoir i to city j.
- The capacity of the piping from reservoir i and city j can handle at most c_{ij} gallons.
- In view of fairness, no city must get more than one third of all its water demand from any single reservoir.

Write a linear program to determine how to supply the water from the reservoirs to the cities, at the lowest cost. (You don't need to solve the LP.)

Solution Create variables g_{ij} representing the number of gallons of water supplied from i to j.

$$\min \sum_{ij} p_{ij} g_{ij}$$

subject to

$$\begin{aligned} \sum_j g_{ij} &\leq G_i && \forall i \\ \sum_i g_{ij} &\geq D_j && \forall j \\ g_{ij} &\leq c_{ij} && \forall (i,j) \\ g_{ij} &\leq D_j/3 && \forall (i,j) \\ g_{ij} &\geq 0 && \forall (i,j) \end{aligned}$$

(b) In a game between players A and B, we have the following payoff matrix:

	Run	Hide
Attack	1	-1
Defend	-1	2

Recall, the row player A wishes to minimize the expected payoff. A plays Attack with probability 0.6 and Defend with probability 0.4.

1. What is the payoff for player B if she runs with probability 0.4 and hides with probability 0.6?
   Solution $0.6\times(0.4\times1+0.6\times-1)+0.4\times(0.4\times-1+0.6\times2) = 0.6\times-0.2+0.4\times0.8 = -0.12+0.32 = 0.2$

2. What is the best (maximum) payoff player B can get?
   Solution The value of Run is $0.6-0.4 = 0.2$ and the value of Hide is $-0.6+2\times0.4 = 0.2$. Since the values of Run and Hide are both 0.2, the best payoff player B can get is 0.2.

Name:	PSU Access ID:

5 Reducing maximum flow *(20 points)*

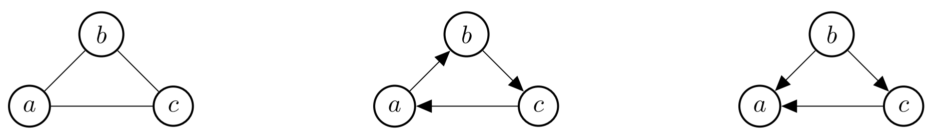
You are given a flow network $G = (V, E)$, with source s and sink t, with all edge capacities equal to 1. You are also given an integer parameter k. Your goal is to delete k edges so as to reduce the maximum $s - t$ flow in G by as much as possible. In other words, you should find a set of edges $F \subseteq E$ so that $|F| = k$ and the maximum $s - t$ flow in $G' = (V, E \setminus F)$ is as small as possible. Give an algorithm to solve this problem with polynomial running time.

Solution This was a homework problem; check the solutions provided in PS1. Since you had seen this problem before, we were more strict while grading the proof of correctness.

Name:	PSU Access ID:

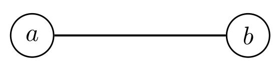
6 Directing *(10 points)*

Given an undirected *connected* graph $G = (V, E)$ we would like to convert G into a directed graph that is strongly connected. We are only allow to choose the direction of each edge of G. For example, for the undirected edge $\{u, v\} \in E$, we can consider either (u, v) or (v, u) for the directed graph. We would like the resulting directed graph to be strongly connected. For example, in the following undirected graph below (on the left), the center figure is an obtainable graph that is strongly connected, and the last one is not.



(a) Give an example of a connected undirected graph where this cannot be done.

Solution



(b) A *cut-edge* in a connected undirected graph is an edge whose removal leaves a graph with two connected components. Given a connected undirected graph G without cut-edges, show how to use the DFS algorithm to convert G into a strongly connected directed graph. Justify the correctness of your approach.

Hint: Consider the type of each edge when deciding in which direction to orient it.

Solution

Algorithm: DFS. Label the tree edges in the explored direction, label the back edges toward the ancestors.

Correctness: The main way to solve this was by contradiction, but there were some interesting inductive solutions. Here, we only present the contradiction way.

By Contradiction: If this fails, consider the sink strongly connected component. Consider the first edge into the component that was explored, explore would not return until the entire component is explored and any other edge incident to this component would then be considered. If the other end was previously explored, DFS would have already explored this edge, thus, DFS would have explored this edge, and it would be directed outward, contradicting the notion that this is the sink component. Thus, there are no other edges incident to this component. Thus, the first edge is a cut-edge, which contradicts the claim that we can not have a cut-edge, so this cannot fail.

Running time: Mention DFS or say "linear time"

Comments

Some students said use the highest post number vertex after doing one DFS. But, since this graph is undirected and connected, the vertex which you start the search will have highest post number (so you're essentially doing DFS twice for no reason).

Also, please do note the difference between what algorithms we can apply to directed graphs and undirected graphs. It doesn't really make much sense to say strongly connected components in undirected graphs (since the graph is connected, we will only get back one connected component). However, this notion does make sense in this problem since we are assigning directions to undirected edges.

One more misconception: A strongly connected component in a directed graph does not imply that there is a cycle going through every single node. However, it is true that every node is part of a cycle, just not necessarily the same one.

Name:	**PSU Access ID:**

Extra space.

Name:	**PSU Access ID:**

Extra space.