CS-3510-C F23 Exam 2 Version B

Vidit Dharmendra Pokharna

TOTAL POINTS

87 / 110

QUESTION 1

Question 1 18 pts

1.1 (i) **0 / 3**

- **- 0 pts** Correct
- ✓ **- 3 pts** *Incorrect*

1.2 (ii) **3 / 3**

- ✓ **- 0 pts** *Correct*
- **- 3 pts** Incorrect

1.3 (iii) **0 / 3**

- **- 0 pts** Correct
- ✓ **- 3 pts** *Incorrect*

1.4 (iv) **3 / 3**

- ✓ **- 0 pts** *Correct*
- **- 3 pts** Incorrect

1.5 (v) **3 / 3**

- ✓ **- 0 pts** *Correct*
- **- 3 pts** Incorrect

1.6 (vi) **3 / 3**

- ✓ **- 0 pts** *Correct*
- **- 3 pts** Incorrect

QUESTION 2

Question 2 15 pts

2.1 (i) **12 / 12**

- ✓ **- 0 pts** *Correct*
- **- 2 pts** One to two edges incorrectly drawn/missing
- **- 4 pts** Three to four edges incorrectly drawn/missing
- **- 5 pts** Five to six edges incorrectly drawn/missing
- **- 6 pts** Edges reveresed in Residual Graph
- **- 9 pts** More than six edges incorrect/missing
- **- 12 pts** Incorrect residual graph

2.2 (ii) **3 / 3**

- ✓ **- 0 pts** *Correct*
- **- 3 pts** Incorrect

QUESTION 3

Question 3 23 pts

3.1 (i) **4 / 4**

- ✓ **- 0 pts** *Correct*
- **- 1 pts** 1-3 numbers incorrect
- **- 2 pts** 4-6 numbers incorrect
- **- 3 pts** 6-9 numbers incorrect
- **- 4 pts** Mostly incorrect/missing

3.2 (ii) **4 / 4**

- ✓ **- 0 pts** *Correct*

- **4 pts** Incorrect

3.3 (iii) **4 / 4**

✓ **- 0 pts** *Correct*

- **2 pts** Added additional source SCC other than A

- **4 pts** Incorrect

3.4 (iv) **4 / 4**

✓ **- 0 pts** *Correct*

- **2 pts** Added additional sink SCC other than BDE

- **4 pts** Incorrect

3.5 (v) **4 / 4**

✓ **- 0 pts** *Correct*

- **2 pts** Claimed no remaining SCCs or additional SCC

- **4 pts** Incorrect

3.6 (vi) **3 / 3**

✓ **- 0 pts** *Correct*

- **3 pts** Incorrect

QUESTION 4

4 Question 4 **15 / 15**

✓ **- 0 pts** *Correct*

- **1 pts** Runtime analysis does not include/incorrect for creating the graph

- **2 pts** Runtime analysis mistake

- **3 pts** Minor error

- **4 pts** Runtime analysis incorrect/missing

- **4 pts** Insufficient justification

- **9 pts** Major error/Mostly incorrect

- **15 pts** Missing/Incorrect

QUESTION 5

5 Question 5 **7 / 9**

- **0 pts** Correct

✓ **- 2 pts** *Minor error*

- **3 pts** Insufficient justification

- **3 pts** Runtime analysis missing/incorrect

- **5 pts** Major error

- **9 pts** Incorrect

1 Need to run Explore on the reversed graph as well

QUESTION 6

Question 6 20 pts

6.1 (i) **0 / 10**

- **0 pts** Correct

- **2 pts** Minor error

- **2 pts** Minor error - Does not use Explore/DFS/BFS correctly (i.e. does not correctly change visited set or construct new graph)

- **3 pts** Insufficient justification

- **3 pts** Runtime analysis missing/incorrect

- **5 pts** Major error

✓ **- 10 pts** *Incorrect*

6.2 (ii) **5 / 10**

- **0 pts** Correct

- **2 pts** Minor error

- **2 pts** Minor error - Does not use Explore/DFS/BFS correctly (i.e. Does not change the visited set or construct a new graph correctly)

- **3 pts** Insufficient justification

- 3 pts Runtime analysis missing/incorrect

✓ **- 5 pts** *Major error*

- 9 pts Incorrect

2 scc wont work

QUESTION 7

7 Question 7 **10 / 10**

✓ **- 0 pts** *Correct*

- 5 pts Correct algorithm, incorrect runtime/analysis missing

- 7 pts Attempts to use Dijkstra but incorrectly

- 7 pts Suboptimal algorithm

- 10 pts fully incorrect, or missing

GTID: 903772087
NAME: Vidit Pokharna

Georgia Institute of Technology

Fall 2023

CS 3510 C – Design & Analysis of Algorithms
Exam 2 Version B

September 28, 2023

TIME ALLOWED: 75 MINS

Name: Vidit Pokharna

GTID: 903772087

GT Username: vpokharna3

INSTRUCTIONS TO STUDENTS

1. Please write your NAME and GTID clearly on all the pages.
2. This examination paper contains **SEVEN (7)** questions and comprises **TEN (10)** printed pages.
3. ONLY write on the front sheets of paper that are numbered. The backs will not be scanned.
4. Calculators are **NOT** allowed.

I am in aware of and the accordance with Academic Honor Code of Georgia Tech and the Georgia Tech Code of Conduct. I'll use no external help on this test. Also, I have read all the instructions on this page.

Signature: viditdpokh

GTID: 903772087
NAME: Vidit Pokharna

Problem 1 *(18 points; 3 points each)*

Indicate whether the following statements are ***true*** or ***false***.

(i) Kruskal's Algorithm will not be correct when run on a graph that contains negative weighted edges.

Answer: True

(ii) The heaviest edge in the graph can belong to a minimum spanning tree.

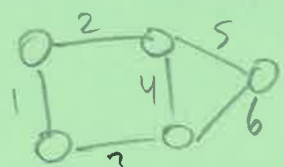
Answer: True

(iii) If you explore from a vertex in a source SCC, then it is not guaranteed that all vertices in the graph are reached.

Answer: False F

(iv) If the weights of edges in an undirected graph G are distinct, there is only one valid MST of G.

Answer: True



(v) Suppose we have a flow network with max flow f, and we multiply all capacities by some constant c. It is not necessarily true that the new max flow of the network is fc.

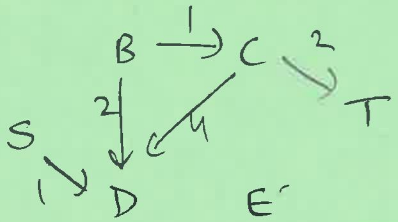
Answer: False F

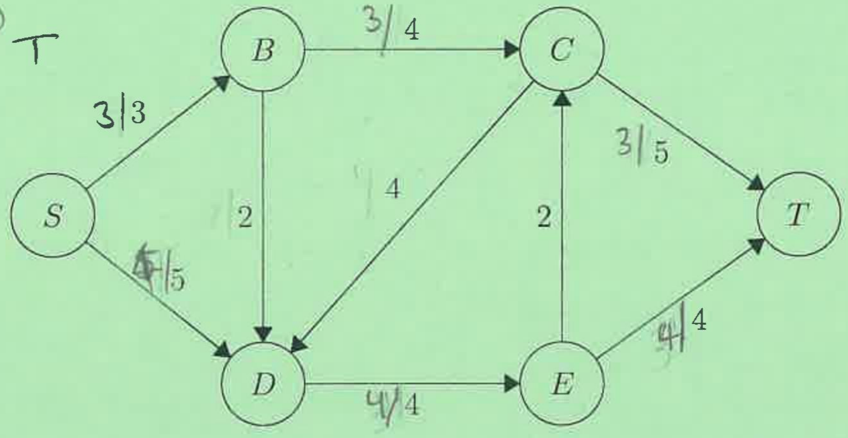
(vi) If all edge capacities in a max flow problem instance are distinct, there may be multiple maximum flow paths.

Answer: True T

GTID: 903772087
NAME: Vidit Pokharna

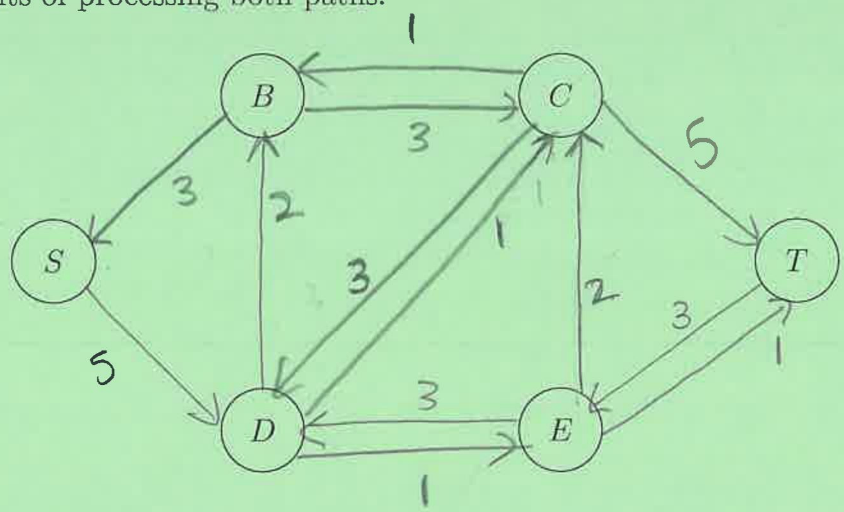
Problem 2 *(15 points)*





Using the above flow network, with each each edge labelled with its capacity, answer the questions below.

(i) (12 points) Run Ford Fulkerson on this graph with two augmenting paths in order: *SBDET*, and *SBCDET*. Complete the residual network below with the results of processing both paths.

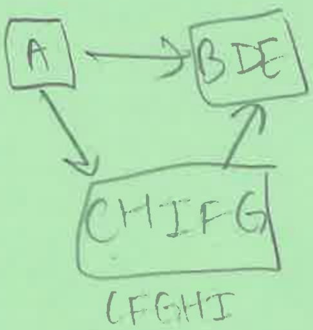


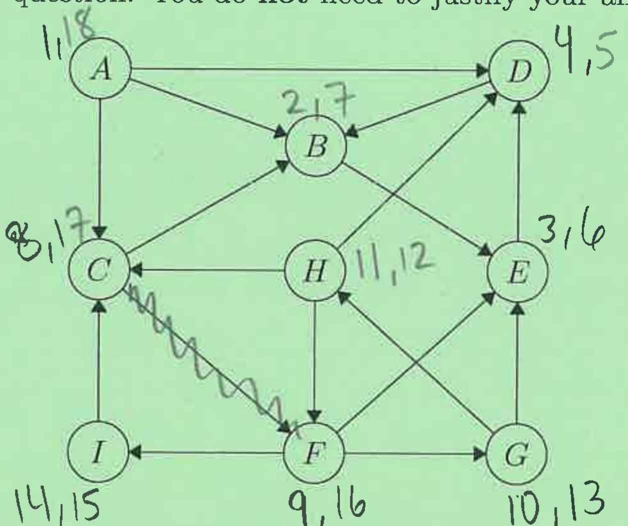
(ii) (3 points) What is the maximum flow from s to t? 7

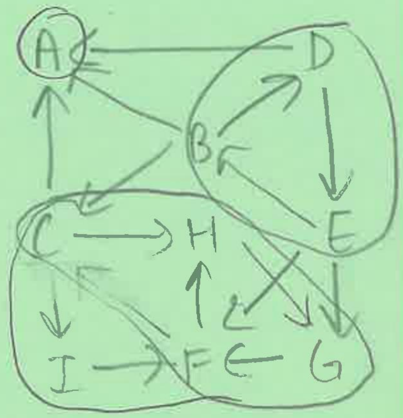
GTID: 903772087
NAME: Vidit Pokharna

Problem 3 *(23 points)*

Answer the questions using the graph G printed below. Write your answers on the line next to the question. You do **not** need to justify your answers.







(i) (4 points) Perform DFS with start vertex A and write the pre and post numbers for each vertex in this graph. Use alphabetical order of the vertices while performing DFS. Start counting pre and post numbers from 1.

A: 1, 18 B: 2, 7 C: 8, 17

D: 4, 5 E: 3, 6 F: 9, 16

G: 10, 13 H: 11, 12 I: 14, 15

(ii) (4 points) How many SCCs are in the graph G?

3

(iii) (4 points) List all the source SCCs separated by commas (if multiple) in this graph G, as group of vertices (list nodes in each SCC in alphabetical order):

A

GTID: 903772087
NAME: Vidit

(iv) (4 points) List all the sink SCCs separated by commas (if multiple) in this graph G, as group of vertices (list nodes in each SCC in alphabetical order):

BDE

(v) (4 points) List all other SCCs separated by commas (if any) in this graph G, as group of vertices (list nodes in each SCC in alphabetical order). If there are no other SCCs, write "N/A":

CFGHI

(vi) (3 points) True or False: Deleting an edge from this graph G can only increase the number of SCCs by at most 1.

False

GTID: 903772087
NAME: Vidit Pokharna

Problem 4 *(15 points)*

Kruskal's
$E \log E$

Suppose you are given an array P which contains n integer coordinates of points on a 2D plane, where $P[i] = (x_i, y_i)$. Define the cost of connecting two points (x_i, y_i) and (x_j, y_j) as the Euclidean distance between them, which is defined as $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

Design an efficient algorithm that computes the minimum cost to connect all n points. **Describe your algorithm in English, justify its correctness, and find its runtime.**

Firstly, we would need to compute the edge lengths between all points. We could run a double for loop iterating through each pair of points, (x_i, y_i) and (x_j, y_j) and find the Euclidean distance ~~between~~ them. For every edge calculated, we can place it in an edge list.

From here, we run kruskals. We would sort the edge list and create an MST from the edge list while implementing union set structure to ensure an acyclic tree. The MST we created would contain the shortest path to connect all n points.

Ensuring we got all edges using the double for loop and making an MST, we can conclude this gives the minimum cost to connect all n points.

6

Creating edges list is $O(n^2)$ and kruskals is $O(n^2 \log(n^2))$, since there is $\frac{n(n-1)}{2}$ edges (factor of n^2). Thus runtime is $O(n^2 \log(n^2))$.

GTID: 903772087
NAME: Vidit Pokharna

Problem 5 *(9 points)*

BFS
DFS

Given a directed graph G, we'll call a vertex v a"happening spot" if, for all other vertices u, there exists a path containing both v and u in G. Give a linear-time algorithm that determines whether a given vertex v is a "happening spot" in G. **Describe your algorithm in English, justify its correctness, and show why it takes linear time.**

we can take in G and v as parameters. We can run DFS on G with starting vertex v. While running, if all vertices are not visited and explore ~~begins~~ another cycle then we know v is not a happening spot. However, if explore runs on all vertices without ending the iteration, the v is a happening spot. thus, running DFS and having a conditional to see whether there are any vertices remaining after one cycle of explore will determine whether v is a happening spot.

DFS runs on O(V+E) and the conditional will check vertex list which is O(V). Therefore this algorithm is O(V+E) which is linear time.

GTID: 903772087
NAME: Vidit Pokharna

Problem 6 *(20 points)*

Suppose we are given an unweighted, undirected graph $G = (V, E)$ where each edge has a blue or red color. You are given vertices $s, t \in G$.

(i) (10 points) Design a linear-time algorithm to determine if there exists a path from s to t that uses blue edges followed only by red edges. **Describe your algorithm in English, justify its correctness, and show why it takes linear time.**

We can reorganize the graph into SCC's. An SCC is only together in this algorithm if all points are of the same color and connected. Once we have found our modified SCC, we have to check if there is a path from the SCC containing s to the SCC containing t. If the SCC's interchange colors in between then false. However if it's only blue SCC's and then red, then true. (can use BFS).

Finding SCC's is O(V+E). Checking for path and seeing if they fit color condition is O(V+E) using BFS. Therefore, this algorithm is O(V+E) which is linear time complexity.

GTID: 903772087
NAME: Vidit Pokharna

(ii) (10 points) Design a linear-time algorithm to determine if there exists a path from s to t that uses alternating edge colors. The path may start with either a red or blue edge. **Describe your algorithm in English, justify its correctness, and show why it takes linear time.**

We can use the same approach as a. Checking SCC's and path from s to t, we can see if there is a path that interchanges color using BFS. If found, we return true. If there is no such path, return false.

This will take O(V+E) time for the same reason as part a.

GTID: 903772087
NAME: Vidit Pokharna

Problem 7 *(Extra Credit; 10 points)*

Suppose we have an undirected weighted graph $G = (V, E, w(e))$, where each weight is a positive integer. Overnight, Donkey Kong erases some of the edge weights without you knowing, that is $w(e) = \bigstar, \forall e \in X$ where $X \subset E$. Once you come back to the graph, you try and "complete" the graph by reassigning positive integer weights to the erased edges such that the length of the shortest path between two given vertices s and t in the resulting graph is exactly L. Given G, X, s, t, and L, design an algorithm that returns a boolean representing whether this is possible. **Design your algorithm in English and show its runtime.**

We can run Djikstra's to check if the shortest path is L or not. If it is L, then return true. If it is greater than L, then return false. Otherwise we can use maxflow mincut to see if there is another path equal to L, where L is max flow.

Djikstra's $= O((V+E)\log V)$

Runtime $= O((V+E)\log V)$, unless we have to use Ford Fulkerson/ max flow min cut

END OF EXAM