Final Exam

ⓘ This is a preview of the published version of the quiz

Started: May 16 at 2:11pm

Quiz Instructions

Question 1

2 pts

Which algorithm can be used to find shortest paths in a weighted, directed acyclic graph in *O(|V|+|E|)* time?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 2

2 pts

Which algorithm is used to find the augmenting paths in the Edmonds-Karp algorithm?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 3

2 pts

Which algorithm can be used to find the maximum matching of a bipartite graph?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 4

2 pts

Which algorithm can be used to identify the connected components of an unweighted, undirected graph in $O(|V|+|E|)$ time?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 5

2 pts

Which algorithm is used when you are processing the sorted edges in Kruskal's algorithm?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 6

2 pts

I have a network of pipes that connect junctions. Each pipe has a length and a diameter. In my control center, I can turn pipes on and off. I am at junction A, and I have been given a chemical that will improve the longevity of each junction. I want to deliver the chemical to each junction. The problem is, that the chemical corrodes the pipes, so I want to minimize the inner surface area of pipes that the chemical touches. What algorithm should I use to determine what pipes to use?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 7

2 pts

Which algorithm is used to find shortest paths in an unweighted, undirected graph?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 8

2 pts

Which algorithm maintains a multimap of nodes, keyed on distance to a starting node?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 9

2 pts

Which algorithm can be used to find the minimum cut of a weighted graph?

Choose your answers from the following -- just enter the number of the answer:

1. Union/Find on Disjoint sets

2. Depth-First Search

3. Breadth-First Search

4. Dijkstra's algorithm for shortest paths

5. Dijkstra's algorithm modified to find paths with minimum flow

6. Dijkstra's algorithm modified to find paths with maximum flow

7. Minimum Spanning Tree

8. Network flow

9. Topological Sort

Question 10

16 pts

We are working on a directed graph defined by the following adjacency matrix:

```
       A   B   C   D   E   F   G   H   I   J
      --- --- --- --- --- --- --- --- --- ---
A |   --  16  60  --  --  --  28  82  65  --
B |   78  --   8  26  29  --  --  30  19  98
C |   --  50  --  --  --  --  --  41  24  --
D |   31  54  12  --  49  96  78  --  44  --
E |   --   5  62  16  --  88  47  --  --  --
F |   59   1  --  --  --  --  38  --  --  91
G |   10  74  20  21  63  --  --   7  --  --
H |   16  --  34  -- 100  97  15  --  --  --
I |   --  74  42  --  --  13  --  --  --  87
J |   48  --  --   2  --  --  82  43  70  --
```

We are running Dijsktra's shortest path algorithm on this graph, starting from node A. When we do this, we process the multimap in the following order:

Node A Distance 0

Node B Distance 16

Node C Distance 24

Node G Distance 28

Node I Distance 35

Node H Distance 35

Node D Distance 42

Node E Distance 45

Node F Distance 48

Node J Distance 114

After we process node A, the distances vector is:

```
                 Distances
  A   B   C   D   E   F   G   H   I   J
  0  16  60   -   -   -  28  82  65   -
```

As we process the nodes after node A, there are four times when we set a distance from '-' to a number, and four times when we change a distance from one number to another. In the four boxes below, enter the nodes whose distances are set, in the order in which they are set. Enter the answer as "Node-distance" -- for example, if your answer is node A with a distance of 5, then answer "A-5".

In the four boxes below, enter the nodes whose distances are changed, in the order in which they are changed. Enter the answer as "Node-newdistance" -- for example, if your answer i1s node A with a new distance of 5, then your answer should be "A-5".

Question 11

16 pts

You are sorting the following string using quicksort, with median-of-three pivot selection:

```
FBRRBABABBB
```

In the very first partition, what is the pivot?

You are sorting the following string using quicksort:

```
MZYTWQIAELBOK
```

Suppose you use the first character, 'M', as a pivot. Please show what the string looks like after you have done the first partition, but before you swap 'M' into place.

You are sorting the following string using quicksort, using the first character, 'H', as the pivot:

```
HBHKHCBCBUHYC
```

Show what the string is after you partition, before you swap 'H' into place.

You are sorting the following string using mergesort:

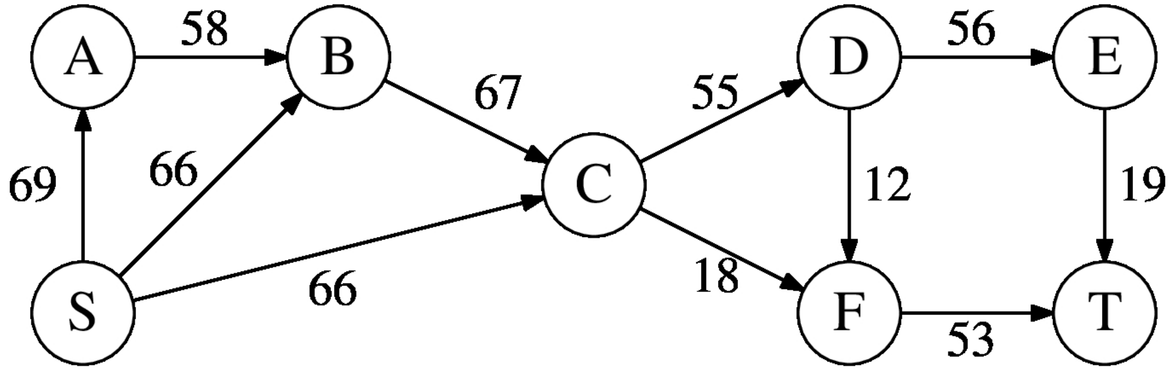
```
FXRIJCAKUWSZ
```

Please enter what the string is before you perform the final merge.

Question 12

16 pts

Helen has the following directed, weighted graph:



Part 1: Helen is using the Edmonds-Karp algorithm for determining the maximum network flow from S to T in this graph. Specify the first augmenting path (Do this by listing the nodes in order in a single string, like SABCDEFG).

Part 2: What is the flow through this path?

Part 3: When Helen processes this path in the residual graph, one edge coming from S has its weight

changed. What is that new weight?

Part 3: One other edge changes weight. What is that edge? (Specify edges as follows -- the edge

from A to B should be specified "AB")

Part 4: And what is its new weight?

Part 5: Additionally, three new edges are added to the residual graph. Name them here:

Part 6: And what is their flow?

Part 7: What is the maximum flow of this graph? (This part will be worth more than the others).

Part 8: What is the minimum cut of this graph -- list the edges and separate them with spaces.

Question 13

14 pts

We have an undirected, weighted graph with eight nodes, A through H. I specify it below by listing the edges and their weights. There are four columns, and each column has the edge and then its weight.

```
GH 10     FG 34     AE 50     CG 73
BC 16     CD 35     AD 56     DE 84
FH 21     DG 40     CE 59     EG 89
CH 23     AF 41     BE 65     AG 92
AC 24     AH 43     CF 69     DH 95
BF 32     EH 47     DF 70     AB 98
BD 33     BH 49     BG 71     EF 99
```

Please enter the minimum spanning tree of this graph. List the edges separated by spaces.

Question 14

16 pts

You are given the following function, that has been written for you:

```
double cost(const string &s1, const string &s2, int n);
```

< You are to write a program calc_min.cpp, that reads the strings *x* and *y* and the integer *z*. You may assume that *x* and *y* are composed of lowercase characters,

and *y* is non-negative. Don't error check. It should return *mincost(x, y, z)*, which is defined as follows:

- *mincost("", "", 0) = 0.*
- Otherwise, *mincost(x, y, z)* is the minimum of:
    - *cost(x, y, z) + mincost(x.substr(1, x.size()-1), y, z)* -- Only if *x* is not "".
    - *cost(x, y, z) + mincost(x.substr(0, x.size()-1), y, z)* -- Only if *x* is not "".
    - *cost(x, y, z) + mincost(x, y.substr(1, y.size()-1), z)* -- Only if *y* is not "".
    - *cost(x, y, z) + mincost(x, y, z-1)* -- Only if *z* is not 0.

Write this program. I know it's going to take a little bit of work. It should be a dynamic program up to "step 2" of dynamic programming. If I were you, I would use a map for my cache that uses strings for keys and doubles for vals.

Comment your program very well, because if I can't understand what you're doing, your comments should help explain so that I may give you partial credit. If you want to use a global variable fot the cache, that's fine. Don't bother with include statements.

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

Question 15

4 pts

In the dynamic programming question, let:

- *X* = *x.size()*.
- *Y* = *y.size()*.
- *Z* = *z*.

In terms of big-O, please tell me how big the cache can be:

Suppose I'm traveling from Needles, California, to Napa, California. I have been able to download a list of all cities between Needles and Napa, plus all of the roads between cities. There are *C* cities and *R* roads. For each road, I have a measure of the road's boringness (which takes into account how long the road is). Napa is northwest of Needles, so I delete all roads that go either south or east. What is the best big-O running time of finding the route that is the least boring, overall.

Saving... Submit Quiz