Test BFS-DFS-MST 2021

Due Oct 21, 2021 at 3:45pm **Points** 60 **Questions** 4
Available Oct 21, 2021 at 2pm - Oct 21, 2021 at 3:50pm about 2 hours
Time Limit 100 Minutes

Instructions

You have maximum 100 minutes for the test, but the deadline is 15:45.

This quiz was locked Oct 21, 2021 at 3:50pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	96 minutes	27.04 out of 60

ⓘ Correct answers are no longer available.

Score for this quiz: **27.04** out of 60
Submitted Oct 21, 2021 at 3:36pm
This attempt took 96 minutes.

Partial

Question 1

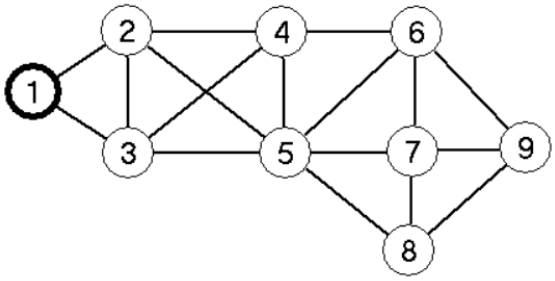
11.79 / 15 pts

BFS (breadth-first search)

Attention! If an answer consists of a sequence of numbers, the numbers must be separated with commas, with no blank. No sign is to be put at the ends of the sequence.
For example: 1,3,4,6

Exercise 1:

Given the graph below.



Run BFS (breadth-first search) on this graph. In all cases, the neighbours of a node must be processed in the increasing order of their indices. Source vertex: **1**

Answer the following questions.

1.a: Consider the first five iterations of the main loop. In each iteration, present the content of the queue at the end of the body of the main loop. The vertices must be separated with commas. Do not use blanks nor other extra characters.

At the end of iteration 1: 2,3

At the end of iteration 2: 3,4,5

At the end of iteration 3: 4,5

At the end of iteration 4: 5,6

At the end of iteration 5: 6,7,8

Consider now the result of BFS.

1.b: What is the maximal *d* value: 4

1.c: Present the *d* and π values of the following vertices. (Write first the *d* value, and then the π value separated by a comma. Use no other character, no blank.)

Vertex 5: 2,2

Vertex 6: 3,4

Vertex 7: 3,5

Vertex 9: 4,6

1.d Let us suppose that we add edge (4,7) to the graph, and we run the algorithm again. Does the *result of the algorithm* change? (Answer *yes* or *no*.) yes

Exercise 2:

Given a digraph, we have run BFS. The parent (i.e. π) values of the vertices:

vertices:	1	2	3	4	5	6	7	8	9	10
π values:	5	4	2	0	2	8	6	4	6	7

Draw the corresponding breadth-first tree and answer the following questions.

2.a: Present the path leading from the source vertex to vertex 1. (Separate the vertices with "," signs. Use no blank.)

The path: 0,4,2,5,1

2.b: Enter the vertices which are at distance 3 from the source. (The vertices must be entered in increasing order according to their indices. They must be separated by commas. Use no blank.)

The vertices at distance 3: 3,5,6

2.c What is the distance of vertex 10: 5

Answer 1:

2,3

Answer 2:

3,4,5

Answer 3:

4,5

Answer 4:

5,6

Answer 5:

6,7,8

Answer 6:

4

Answer 7:

2,2

Answer 8:

3,4

Answer 9:

3,5

Answer 10:

4,6

Answer 11:

yes

Answer 12:

0,4,2,5,1

Answer 13:

3,5,6

Answer 14:

5

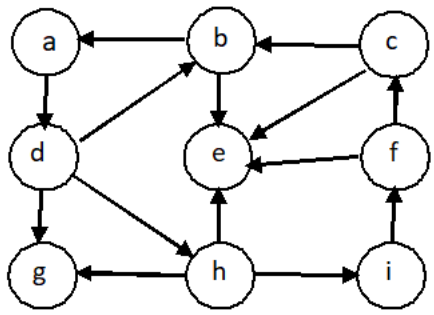
Partial

Question 2

6.25 / 15 pts

DFS (depth-first search)

Given the following graph:



Run DFS on this graph. Use the alphabetical convention. Classify the edges of the graph.

Answer the following questions.

a) ***Enter the** d **and** f **values of the following vertices.***
(Write first the *d* value, and then the *f* value separated by a comma. Use no other character, no blank.)

Vertex **c**: 5,8

Vertex **e**: 6,7

b) Provide the classification of the following edges entering the appropriate letter.
(T -- Tree edge, F -- Forward edge, C -- Cross edge, B -- Back edge)

Edge **(b,a)** : F

Edge **(b,e)**: T

Edge **(h,e)**: F

c) The number of forward edges = 8

d) The number of trees of the depth-first forest = 6

e) Is the original graph a DAG? (Y - yes, N - no) Y

f) ***Consider the subgraph containing vertices b,c,e,f and the edges between them. Enter the topological order of the vertices computed by DFS topological sort on this subgraph. Use the alphabetical convention in DFS.***

f , c , b ,

e

Answer 1:

5,8

Answer 2:

6,7

Answer 3:

F

Answer 4:

T

Answer 5:

F

Answer 6:

8

Answer 7:

6

Answer 8:

Y

Answer 9:

f

Answer 10:

c

Answer 11:

b

Answer 12:

e

Partial

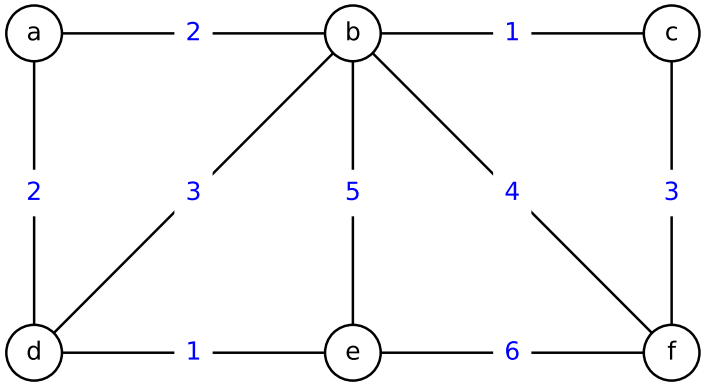
Question 3

3 / 15 pts

Algorithm of Prim:

Attention! If an answer consists of a sequence of numbers or letters, the elements of the sequence must be separated with commas, with no blank. No sign is to be put at the ends of the sequence.

Illustrate the algorithm of Prim on the graph below as you have seen it in the classroom where the *root* is vertex *f*. Then answer the following questions.



1. Give the final *c* values of the vertices 3,1,2,2,1 separated by commas, and then the *p* values of the vertices 3,4,6,8,9 similarly (pointer null must be written as 0).
2. Give the sequence numbers of the iterations of the main loop of the algorithm where the *c* value of vertex *e* is changed. 5
3. Give the different values of *p(e)* according to the chronological order where each *p(e)* value must be given only once. 3,4,6,8,9
4. Give the preorder traversal of the MST calculated. The vertices must be separated with commas. (The root is vertex *f*. The children of a parent are processed according to the increasing weights of the edges leading to them. In case of equal weights use the alphabetical order.) f,c,b,a,d,e

Answer 1:

3,1,2,2,1

Answer 2:

3,4,6,8,9

Answer 3:

5

Answer 4:

3,4,6,8,9

Answer 5:

f,c,b,a,d,e

Partial

Question 4

6 / 15 pts

Given an undirected connected acyclic graph, i.e. an undirected tree. The tree is given in adjacency list representation using array *A/1* of *n* elements. (The elements of the array are pointers identifying S1Ls, i.e. simple one-way lists.)

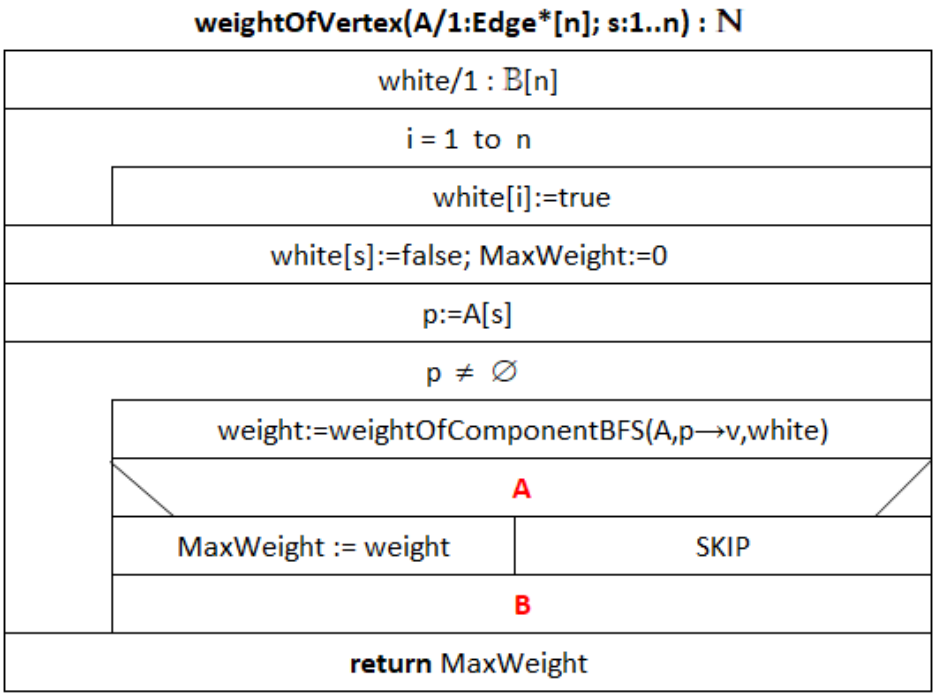
Clearly, if we remove vertex *s* of this tree (together with the edges of it), we receive a forest of undirected trees. The size of a tree is the number of vertices in this tree. Let the weight of vertex *s* be the size of the biggest tree of this forest.

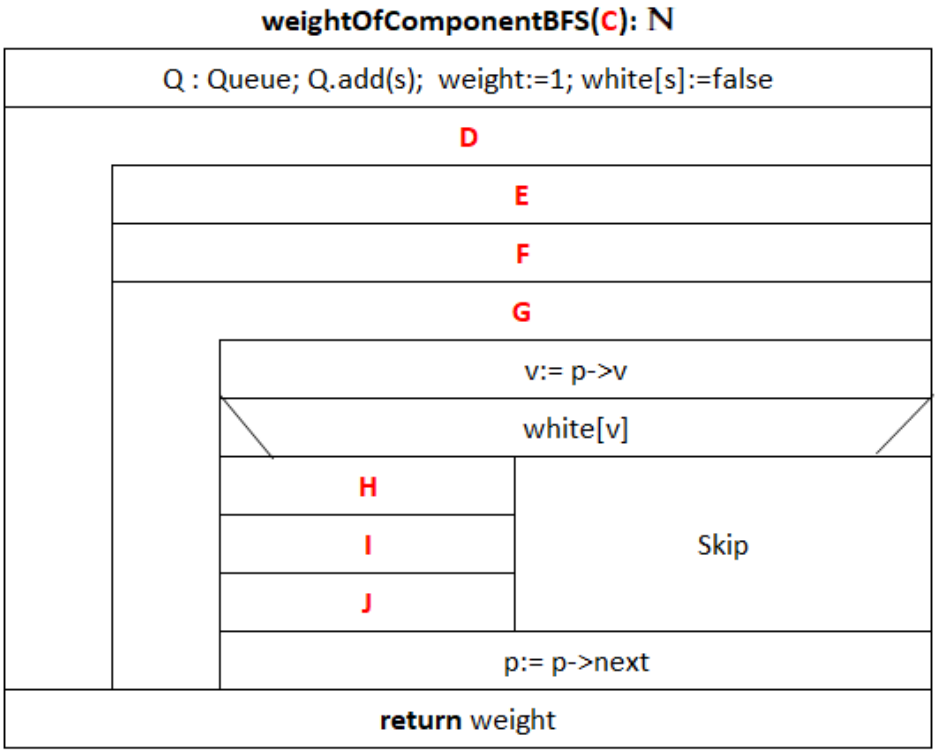
Write an algorithm based on BFS (breadth-first search) which computes the weight of vertex s of the original tree. Operational complexity: $O(n + m)$ where *m* is the number of edges of the graph.

Select the appropriate code segments from the drop-down lists.

A: [Select]

B: [Select]





C: [Select]

D: [Select]

E: u:=Q.top()

F: [Select]

G: [Select]

H: [Select]

I: [Select]

J: [Select]

Answer 1:

weight > MaxWeight

Answer 2:

p:= A[i]->next

Answer 3:

A/1:Edge*[n]; s:1..n; white:B[n]

Answer 4:

Q.isEmpty()

Answer 5:

u:=Q.top()

Answer 6:

p:= A[i]->next

Answer 7:

p->next ≠ 0

Answer 8:

weight:=weight+1

Answer 9:

white[v]:=false

Answer 10:

Q.add(v)

Quiz Score: **27.04** out of 60