ELL781: Software Fundamentals for Computer Technology

Major Test, Form: A (**please write this Form ID on the cover page of your answer script**)

Maximum marks: 24

Section 1. Multiple choice questions

Each question may have any number of correct choices. List (in clear handwriting) the letters corresponding to all choices you believe to be correct (1 mark for each correct choice, -1/2 for each incorrect choice). No justification is required.

1. Consider a flow network on which you obtain a flow f using the Edmonds-Karp algorithm. Suppose I give you a randomly chosen cut (S, T) of the same network. Which of the following must be true?
   (a) $|f| = c(S, T)$
   (b) $|f| \leq c(S, T)$
   (c) $|f| \geq c(S, T)$
   (d) $|f| < c(S, T)$
   (e) $|f| > c(S, T)$

2. Consider an undirected graph with 10 vertices and 15 edges (as usual, there are no self-loops), such that no vertex has degree greater than 3. What is the smallest possible number of vertices in a minimum vertex cover of this graph?
   (a) 9
   (b) 3
   (c) 4
   (d) 5
   (e) 6

3. Consider an undirected graph with 7 vertices and 20 edges (as usual, there are no self-loops). Can you say something about the minimum number of colours needed for a valid colouring of this graph? Pick the most precise choice.
   (a) It is exactly 6
   (b) It is at least 6
   (c) It is exactly 7
   (d) It is at least 5

4. You are given a large network (graph) consisting of data from Facebook: a million vertices corresponding to users, and undirected edges corresponding to friendships between users. Each edge is weighted in inverse proportion to the frequency of interaction between the two friends. You need to find a Minimum Spanning Tree of this network. Which algorithm will be better?
   (a) Prim's algorithm should be faster.
   (b) Kruskal's algorithm should be faster.
   (c) Both algorithms should take about the same time.

5. Consider the problem of determining whether or not there exists a path of length $\leq k$ between *any* pair of vertices in a directed graph. Which complexity class(es) does this problem belong to (assuming $P \neq NP$)?

   (a) P

   (b) NP

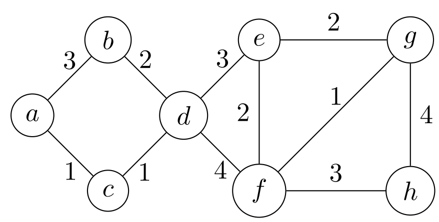
   (c) NP-complete

   (d) NP-hard

Section 2. Short Answer Questions

NB: Please read all questions carefully. There may be subtle differences between what the question is asking for and the context in which things have been discussed during the lectures, and these have to be taken into account whilst answering. Working/derivation for all your answers should be shown fully and clearly.

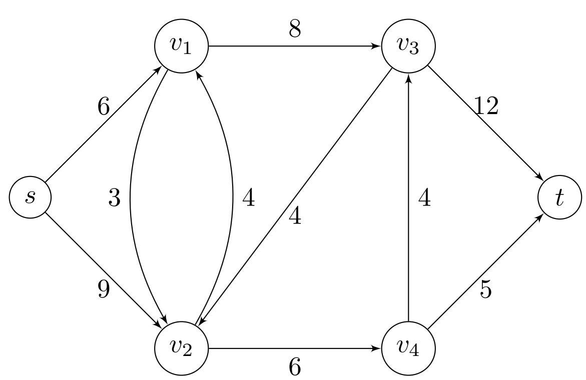
6. Consider the below graph.



(a) Consider the edge (d, e). Can you prove that there exists an MST of the given graph which includes this edge? (Hint: Use the MST property. You need not prove the property itself, but should give a formal and precise argument showing how the property can be applied here to prove the specific claim to be established.) **[1.5]**

(b) Show clearly the *sequence* in which edges will get added to the constructed MST, when you execute Kruskal's algorithm on the above graph. Wherever there is a choice, assume edges will be selected in alphabetical order as per the vertex labelling. **[1.5]**

7. Consider the below flow network.



(a) Show the execution of the Edmonds-Karp algorithm (*i.e.*, Ford-Fulkerson with BFS) on the above network, clearly depicting all the steps (you should draw for each iteration the augmenting path found, the resulting augmented flow, and the resulting residual network). Assume that for each vertex, BFS will always scan through its neighbours in the given numerical sequence, with s at the start and t at the end. **[5]**

(b) In the maximum flow obtained above, what is the flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$? What is the capacity of this cut? **[1]**

(c) Is the cut just examined a minimum cut of the given flow network? If not, find a minimum cut. **[1]**

8. Recall the algorithm APPROX-VERTEX-COVER, which maintains a set of uncovered edges, and at each iteration picks one edge from this set at random and adds both its endpoints to the cover set.

   Suppose a graph has a minimum vertex cover with an odd number of vertices. Clearly, for such graphs, APPROX-VERTEX-COVER can never return the optimal solution, as it always adds two nodes to the cover set at a time and hence always returns an even-sized cover.

   But now consider the class of all graphs for which the minimum vertex cover has an even number of vertices. Within this class, do there exist graphs for which APPROX-VERTEX-COVER can never give the optimal solution, no matter what sequence of edges it may happen to pick?

   If yes, give an example of such a graph and explain/prove why APPROX-VERTEX-COVER can never be optimal on it. If no, explain/prove why APPROX-VERTEX-COVER will always be able to find an optimal cover for such graphs, if it happens to pick the right edge sequence. **[3]**

9. You are visiting the annual gathering of the Taureans Society: a group of people who all have the star sign Taurus (*i.e.*, their birthday falls during 19th April to 20th May). How many minimum people should there be at the gathering so that the expected number of shared birthdays amongst them is at least 1? **[4]**

ELL781: Software Fundamentals for Computer Technology

Major Test, Form: B (**please write this Form ID on the cover page of your answer script**)
Maximum marks: 24

Section 1. Multiple choice questions

Each question may have any number of correct choices. List (in clear handwriting) the letters corresponding to all choices you believe to be correct (1 mark for each correct choice, -1/2 for each incorrect choice). No justification is required.

1. You are given a large network (graph) consisting of data from Facebook: a million vertices corresponding to users, and undirected edges corresponding to friendships between users. Each edge is weighted in inverse proportion to the frequency of interaction between the two friends. You need to find a Minimum Spanning Tree of this network. Which algorithm will be better?

   (a) Kruskal's algorithm should be faster.
   (b) Both algorithms should take about the same time.
   (c) Prim's algorithm should be faster.

2. Consider an undirected graph with 7 vertices and 20 edges (as usual, there are no self-loops). Can you say something about the minimum number of colours needed for a valid colouring of this graph? Pick the most precise choice.

   (a) It is at least 5
   (b) It is exactly 6
   (c) It is at least 6
   (d) It is exactly 7

3. Consider the problem of determining whether or not there exists a path of length $\leq k$ between *any* pair of vertices in a directed graph. Which complexity class(es) does this problem belong to (assuming $P \neq NP$)?

   (a) NP
   (b) NP-complete
   (c) NP-hard
   (d) P

4. Consider an undirected graph with 10 vertices and 15 edges (as usual, there are no self-loops), such that no vertex has degree greater than 3. What is the smallest possible number of vertices in a minimum vertex cover of this graph?

   (a) 3
   (b) 4
   (c) 5
   (d) 6
   (e) 9

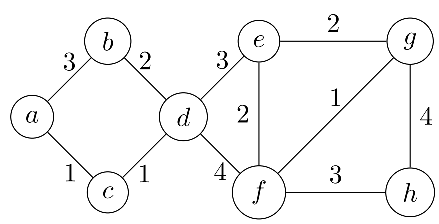
5. Consider a flow network on which you obtain a flow f using the Edmonds-Karp algorithm. Suppose I give you a randomly chosen cut (S, T) of the same network. Which of the following must be true?

   (a) $|f| > c(S, T)$
   (b) $|f| = c(S, T)$
   (c) $|f| \leq c(S, T)$
   (d) $|f| \geq c(S, T)$
   (e) $|f| < c(S, T)$

Section 2. Short Answer Questions

NB: Please read all questions carefully. There may be subtle differences between what the question is asking for and the context in which things have been discussed during the lectures, and these have to be taken into account whilst answering. Working/derivation for all your answers should be shown fully and clearly.

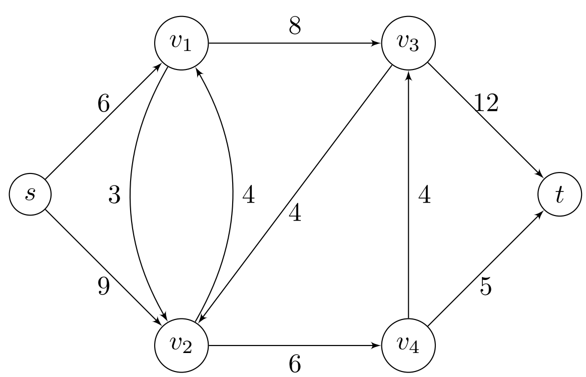
6. Consider the below graph.



(a) Consider the edge (d, e). Can you prove that there exists an MST of the given graph which includes this edge? (Hint: Use the MST property. You need not prove the property itself, but should give a formal and precise argument showing how the property can be applied here to prove the specific claim to be established.) **[1.5]**

(b) Show clearly the *sequence* in which edges will get added to the constructed MST, when you execute Kruskal's algorithm on the above graph. Wherever there is a choice, assume edges will be selected in alphabetical order as per the vertex labelling. **[1.5]**

7. Consider the below flow network.



(a) Show the execution of the Edmonds-Karp algorithm (*i.e.*, Ford-Fulkerson with BFS) on the above network, clearly depicting all the steps (you should draw for each iteration the augmenting path found, the resulting augmented flow, and the resulting residual network). Assume that for each vertex, BFS will always scan through its neighbours in the given numerical sequence, with s at the start and t at the end. **[5]**

(b) In the maximum flow obtained above, what is the flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$? What is the capacity of this cut? **[1]**

(c) Is the cut just examined a minimum cut of the given flow network? If not, find a minimum cut. **[1]**

8. Recall the algorithm APPROX-VERTEX-COVER, which maintains a set of uncovered edges, and at each iteration picks one edge from this set at random and adds both its endpoints to the cover set.

   Suppose a graph has a minimum vertex cover with an odd number of vertices. Clearly, for such graphs, APPROX-VERTEX-COVER can never return the optimal solution, as it always adds two nodes to the cover set at a time and hence always returns an even-sized cover.

   But now consider the class of all graphs for which the minimum vertex cover has an even number of vertices. Within this class, do there exist graphs for which APPROX-VERTEX-COVER can never give the optimal solution, no matter what sequence of edges it may happen to pick?

   If yes, give an example of such a graph and explain/prove why APPROX-VERTEX-COVER can never be optimal on it. If no, explain/prove why APPROX-VERTEX-COVER will always be able to find an optimal cover for such graphs, if it happens to pick the right edge sequence. **[3]**

9. You are visiting the annual gathering of the Taureans Society: a group of people who all have the star sign Taurus (*i.e.*, their birthday falls during 19th April to 20th May). How many minimum people should there be at the gathering so that the expected number of shared birthdays amongst them is at least 1? **[4]**

ELL781: Software Fundamentals for Computer Technology

Major Test, Form: C (**please write this Form ID on the cover page of your answer script**)

Maximum marks: 24

Section 1. Multiple choice questions

Each question may have any number of correct choices. List (in clear handwriting) the letters corresponding to all choices you believe to be correct (1 mark for each correct choice, -1/2 for each incorrect choice). No justification is required.

1. Consider an undirected graph with 10 vertices and 15 edges (as usual, there are no self-loops), such that no vertex has degree greater than 3. What is the smallest possible number of vertices in a minimum vertex cover of this graph?

   (a) 4
   (b) 5
   (c) 6
   (d) 9
   (e) 3

2. Consider the problem of determining whether or not there exists a path of length $\leq k$ between *any* pair of vertices in a directed graph. Which complexity class(es) does this problem belong to (assuming $P \neq NP$)?

   (a) NP-complete
   (b) NP-hard
   (c) P
   (d) NP

3. You are given a large network (graph) consisting of data from Facebook: a million vertices corresponding to users, and undirected edges corresponding to friendships between users. Each edge is weighted in inverse proportion to the frequency of interaction between the two friends. You need to find a Minimum Spanning Tree of this network. Which algorithm will be better?

   (a) Both algorithms should take about the same time.
   (b) Prim's algorithm should be faster.
   (c) Kruskal's algorithm should be faster.

4. Consider a flow network on which you obtain a flow f using the Edmonds-Karp algorithm. Suppose I give you a randomly chosen cut (S, T) of the same network. Which of the following must be true?

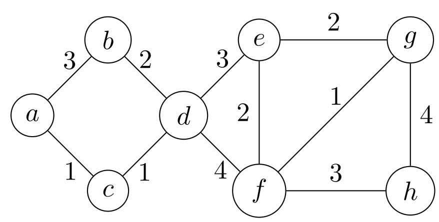
   (a) $|f| < c(S, T)$
   (b) $|f| > c(S, T)$
   (c) $|f| = c(S, T)$
   (d) $|f| \leq c(S, T)$
   (e) $|f| \geq c(S, T)$

5. Consider an undirected graph with 7 vertices and 20 edges (as usual, there are no self-loops). Can you say something about the minimum number of colours needed for a valid colouring of this graph? Pick the most precise choice.
   (a) It is exactly 7
   (b) It is at least 5
   (c) It is exactly 6
   (d) It is at least 6

Section 2. Short Answer Questions

NB: Please read all questions carefully. There may be subtle differences between what the question is asking for and the context in which things have been discussed during the lectures, and these have to be taken into account whilst answering. Working/derivation for all your answers should be shown fully and clearly.

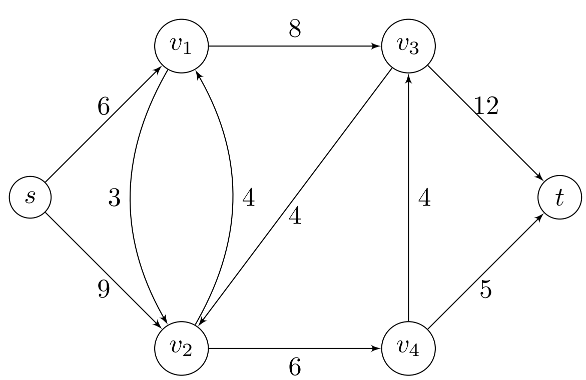
6. Consider the below graph.



(a) Consider the edge (d, e). Can you prove that there exists an MST of the given graph which includes this edge? (Hint: Use the MST property. You need not prove the property itself, but should give a formal and precise argument showing how the property can be applied here to prove the specific claim to be established.) **[1.5]**

(b) Show clearly the *sequence* in which edges will get added to the constructed MST, when you execute Kruskal's algorithm on the above graph. Wherever there is a choice, assume edges will be selected in alphabetical order as per the vertex labelling. **[1.5]**

7. Consider the below flow network.



(a) Show the execution of the Edmonds-Karp algorithm (*i.e.*, Ford-Fulkerson with BFS) on the above network, clearly depicting all the steps (you should draw for each iteration the augmenting path found, the resulting augmented flow, and the resulting residual network). Assume that for each vertex, BFS will always scan through its neighbours in the given numerical sequence, with s at the start and t at the end. **[5]**

(b) In the maximum flow obtained above, what is the flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$? What is the capacity of this cut? **[1]**

(c) Is the cut just examined a minimum cut of the given flow network? If not, find a minimum cut. **[1]**

8. Recall the algorithm APPROX-VERTEX-COVER, which maintains a set of uncovered edges, and at each iteration picks one edge from this set at random and adds both its endpoints to the cover set.

   Suppose a graph has a minimum vertex cover with an odd number of vertices. Clearly, for such graphs, APPROX-VERTEX-COVER can never return the optimal solution, as it always adds two nodes to the cover set at a time and hence always returns an even-sized cover.

   But now consider the class of all graphs for which the minimum vertex cover has an even number of vertices. Within this class, do there exist graphs for which APPROX-VERTEX-COVER can never give the optimal solution, no matter what sequence of edges it may happen to pick?

   If yes, give an example of such a graph and explain/prove why APPROX-VERTEX-COVER can never be optimal on it. If no, explain/prove why APPROX-VERTEX-COVER will always be able to find an optimal cover for such graphs, if it happens to pick the right edge sequence. **[3]**

9. You are visiting the annual gathering of the Taureans Society: a group of people who all have the star sign Taurus (*i.e.*, their birthday falls during 19th April to 20th May). How many minimum people should there be at the gathering so that the expected number of shared birthdays amongst them is at least 1? **[4]**

ELL781: Software Fundamentals for Computer Technology

Major Test, Form: D (**please write this Form ID on the cover page of your answer script**)

Maximum marks: 24

Section 1. Multiple choice questions

Each question may have any number of correct choices. List (in clear handwriting) the letters corresponding to all choices you believe to be correct (1 mark for each correct choice, -1/2 for each incorrect choice). No justification is required.

1. Consider an undirected graph with 7 vertices and 20 edges (as usual, there are no self-loops). Can you say something about the minimum number of colours needed for a valid colouring of this graph? Pick the most precise choice.
   (a) It is at least 6
   (b) It is exactly 7
   (c) It is at least 5
   (d) It is exactly 6
2. You are given a large network (graph) consisting of data from Facebook: a million vertices corresponding to users, and undirected edges corresponding to friendships between users. Each edge is weighted in inverse proportion to the frequency of interaction between the two friends. You need to find a Minimum Spanning Tree of this network. Which algorithm will be better?
   (a) Kruskal's algorithm should be faster.
   (b) Prim's algorithm should be faster.
   (c) Both algorithms should take about the same time.
3. Consider an undirected graph with 10 vertices and 15 edges (as usual, there are no self-loops), such that no vertex has degree greater than 3. What is the smallest possible number of vertices in a minimum vertex cover of this graph?
   (a) 5
   (b) 6
   (c) 9
   (d) 3
   (e) 4
4. Consider the problem of determining whether or not there exists a path of length $\leq k$ between *any* pair of vertices in a directed graph. Which complexity class(es) does this problem belong to (assuming $P \neq NP$)?
   (a) NP-hard
   (b) P
   (c) NP
   (d) NP-complete

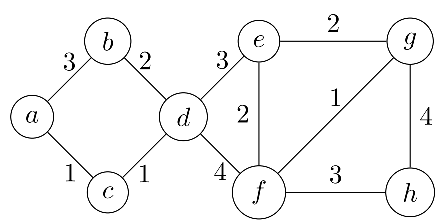
5. Consider a flow network on which you obtain a flow f using the Edmonds-Karp algorithm. Suppose I give you a randomly chosen cut (S, T) of the same network. Which of the following must be true?

   (a) $|f| \geq c(S, T)$
   (b) $|f| < c(S, T)$
   (c) $|f| > c(S, T)$
   (d) $|f| = c(S, T)$
   (e) $|f| \leq c(S, T)$

Section 2. Short Answer Questions

NB: Please read all questions carefully. There may be subtle differences between what the question is asking for and the context in which things have been discussed during the lectures, and these have to be taken into account whilst answering. Working/derivation for all your answers should be shown fully and clearly.

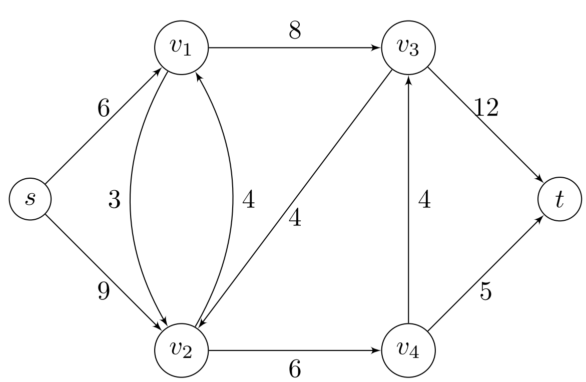
6. Consider the below graph.



   (a) Consider the edge (d, e). Can you prove that there exists an MST of the given graph which includes this edge? (Hint: Use the MST property. You need not prove the property itself, but should give a formal and precise argument showing how the property can be applied here to prove the specific claim to be established.) **[1.5]**

   (b) Show clearly the *sequence* in which edges will get added to the constructed MST, when you execute Kruskal's algorithm on the above graph. Wherever there is a choice, assume edges will be selected in alphabetical order as per the vertex labelling. **[1.5]**

7. Consider the below flow network.



   (a) Show the execution of the Edmonds-Karp algorithm (*i.e.*, Ford-Fulkerson with BFS) on the above network, clearly depicting all the steps (you should draw for each iteration the augmenting path found, the resulting augmented flow, and the resulting residual network). Assume that for each vertex, BFS will always scan through its neighbours in the given numerical sequence, with s at the start and t at the end. **[5]**

(b) In the maximum flow obtained above, what is the flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$? What is the capacity of this cut? **[1]**

(c) Is the cut just examined a minimum cut of the given flow network? If not, find a minimum cut. **[1]**

8. Recall the algorithm APPROX-VERTEX-COVER, which maintains a set of uncovered edges, and at each iteration picks one edge from this set at random and adds both its endpoints to the cover set.

   Suppose a graph has a minimum vertex cover with an odd number of vertices. Clearly, for such graphs, APPROX-VERTEX-COVER can never return the optimal solution, as it always adds two nodes to the cover set at a time and hence always returns an even-sized cover.

   But now consider the class of all graphs for which the minimum vertex cover has an even number of vertices. Within this class, do there exist graphs for which APPROX-VERTEX-COVER can never give the optimal solution, no matter what sequence of edges it may happen to pick?

   If yes, give an example of such a graph and explain/prove why APPROX-VERTEX-COVER can never be optimal on it. If no, explain/prove why APPROX-VERTEX-COVER will always be able to find an optimal cover for such graphs, if it happens to pick the right edge sequence. **[3]**

9. You are visiting the annual gathering of the Taureans Society: a group of people who all have the star sign Taurus (*i.e.*, their birthday falls during 19th April to 20th May). How many minimum people should there be at the gathering so that the expected number of shared birthdays amongst them is at least 1? **[4]**