Algorithm Final Exam

12-20-2018

Prof Nair.

Part III: [6 + 2 + 2]

(a) **(to be illustrated step by step)** Heap sort [7, 2, 9, 1, 5, 4, 3, 8, 6] in **ascending** order using **in-place bottom-up iterative** method. In this case, heap is maintained in an array as explained in your class notes.

(b) What is the time complexity to build a heap using in-place bottom-up iterative method.

(c) Prove the result in (b).

PART I: [1 + 1 + 1 + 1 + 1 + 1]

(a) What is the relationship between the number of vertices and the number of edges of

(b) Prove the result you stated in part (a).

(c) What is the relationship between odd cycles and bipartite graphs?

(d) Prove the result you stated in part (c).

(e) Prove that if G is disconnected, then G^c is connected.

(f) Draw a graph G with 5 or more vertices such that both G and G^c are connected. If su

doesn't exists, please write "G doesn't exist".

Part II: [1 + 1 + 1 + 2 + 5]

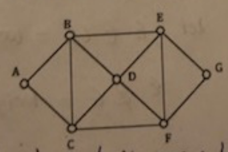
(a) What are the properties of a Red-Black tree.

(b) True or False: Number of Red nodes <= number of Black nodes.

(c) True or False:The time complexity to build a n node Red-Black is O(n).

(d) Write a nondeterministic algorithm to search an "item" in an integer array. What is its complexity?

(e) Illustrate the proof that the HamiltonianCycle problem is polynomial reducible to TSP considering the following Hamiltonian graph—an instance of HamiltonianCycle—and transforming it to a TSP instance in polynomial time so that **a solution to the HC probl** **a solution to the TSP problem, and conversely.**



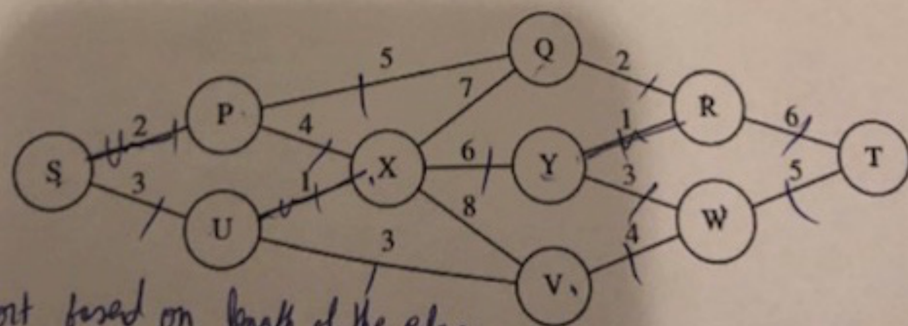
black

Part IV: [6 + 6]

(a) **(to be illustrated step by step)** Compute shortest path from **A** to **F** based on the adjacency matrix given below. Show all steps.

	A	B	C	D	E	F
A	0	4	5	0	0	0
B	0	0	-1	-2	4	0
C	0	0	0	0	-3	0
D	0	0	3	0	6	5
E	0	0	0	0	0	4
F	0	0	0	0	0	0

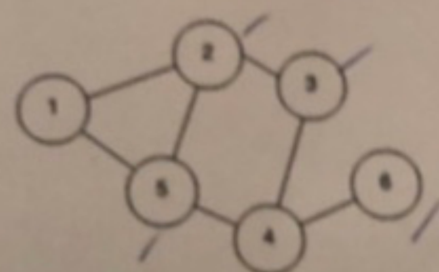
(b) **(To be illustrated step by step)** Apply Kruskal's algorithm to the following graph to compute minimum spanning tree. Please show the minimum spanning tree. What is the length of the minimum spanning tree.



first sort based on length of the edges.

... (S,U) (U,V) (Y,W), (V,W), (

Part VI: [2 + 2 + 2] Consider the following graph G = (V, E).



(a) Give a spanning tree.
(b) Give a spanning subgraph that is not a tree.
(c) Let W = {2, 3, 5, 6}. What is G[W] (G[W] is the subgraph induced by

Part V: [3 + 3]

(a) What is the minimum number of edges required to guarantee that a graph on n vert connected (irrespective of which edges are present).
(b) Prove the result (a).

a)