1. [20 points] TRUE/FALSE OR PICK ONE. *No need for justification.* −4

(b) TRUE/FALSE
The maximum spanning tree (the spanning tree of largest total weight) of an undirected,

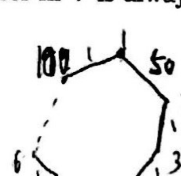
(e) TRUE/FALSE
Prim's algorithm for computing minimum spanning trees gives correct result even if the input graph has negative edge weights.

True.

(f) TRUE/FALSE
Even if all edge capacities in a flow network are prime numbers, the maximum flow value for this network can be a non-prime number.

True. 3+5

(g) TRUE/FALSE
Let $G = (V, E)$ be a connected, undirected graph with a weight function $w : E \to \mathbb{N}$ defined on its edges. The shortest path between any two vertices in V is always part of some minimum spanning tree of G.



MST = 99.
(1,2) ∉ MST
(1,2) ∈ (1,2)

(h) PICK ONE

(ii) $\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} \Phi(D_i)$

(iii) $\Phi(D_i) \geq \Phi(D_{i-1})$

(iii) $d_{ij}^{(k)} = \min_{1 \leq t \leq n}\{d_{it}^{(k-1)} + a_{tj}\}$ for $k \geq 1$ and $d_{ij}^{(0)} = 0$

(iv) $d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)} + a_{ij},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}$ for $k \geq 1$ and $d_{ij}^{(0)} = 0$

(j) PICK ONE

Let $f_1, f_2 : V \times V \to R$ be two different flows on a flow network $G = (V, E)$ with a capacity $c : V \times V \to R^+ \cup \{0\}$. Then, $f_1 + f_2$, which is defined as $(f_1 + f_2)(e) = f_1(e) + f_2(e)$ for every $e \in V \times V$, is NOT necessarily a flow in G because it can violate ___.

(i) the capacity constraint

(ii) the conservation law

(iii) the skew symmetry property

(iv) all of the above

2. [10 points] PROVING YOUR CLAIM

either 3 or 5 which is a prime number.

Every time we could put one vertex out of our consideration.
So it will be finished by $O(V)$ not $O(|V|\log|V|)$.

15

3. [15 points] AMORTIZED ANALYSIS (AGGREGATE OR POTENTIAL)
Consider a street in Jacksonville orthogonal to Atlantic Ocean shore. Only some of the

Answer: we use the potential method, and we set the function as:

$\Phi = 2 \times$ height of the stack.

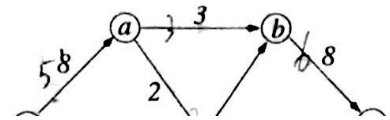
② S is not empty and we need ... we popped k elements until we find an element bigger than him, assuming we popped k elements.

$$\hat{C} = 1 + k \times 2 + \Phi' - \Phi = (2k+1) + 2(-k+1) = 3.$$

So, although it seems we need to do many jobs if we need to come to situation ②, but we see the amortized cost is still const.

In conclusion, the amortized cost for using this algorithm for $A[n]$ which contains n elements, is $O(n)$.

4. [30 points] **NETWORK FLOW**



and < s, c, d, t >. Draw the residual graph for this flow assignment of value 6.

(c) Give a list of additional (to part (b)) augmentations that results in a maximum flow.

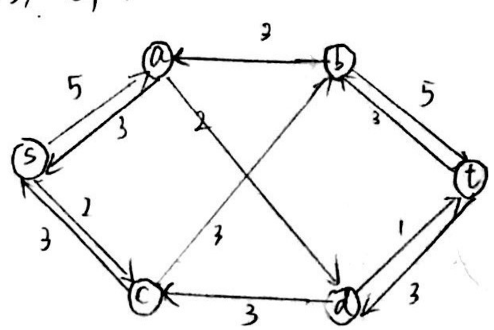
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

T = {b, c, d, t}, where c = 10. 5

(b) Gf :



5

The residual capacity of each augmenting path is 3. so the total value now is 6.

Chose the path: <s, a, d, t> with value 1.

Finally, chose the path: <s, a, d, c, b, t> with value 1.

we got the Maximum Flow.

so the list of AP is: 5

< s, c, b, t >, < s, a, d, t >,

< s, a, d, c, b, t >.

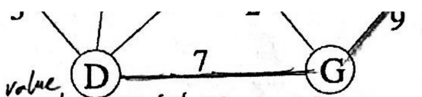
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

So, if the Maximum flow increased, then it must be [illegible]
edge should cross the minimal cut (S, T).
So as we find the minimal cut in (a) as ({s, a}, {b, c, d, t}), so we first need to
find an edge which doesn't exist in Gf and connect S to T.
So, the bottleneck edge might be (s, c), (a, d) and (a, b), draw the Gf for the
3 choices, we find only increasing (a, b) will induce an Augmenting path. 5
So it's (a, b)

5. [20 points] SPANNING TREE & FEEDBACK EDGES

4	d	0	4	1	3	6	10	7	E	AD, AB, BC, DE	EG
5	d	0	4	1	3	6	10	2	G	AD, AB, BC, DE, EG	GF
6	d	0	4	1	3	6	9	2	F	AD, AB, BC, DE, EG	-

the modified graph, starting with vertex A.



Modified: Every time, choose the largest edge. Only get relaxed when there is an edge weight larger than it's value kept (except ∞).

Step		A	B	C	D	E	F	G	Selected Vertex	MaxST Edge	Relaxed Edges
0	d	0	∞	∞	∞	∞	∞	∞	A	-	AB, AD
1	d	0	4	∞	3	∞	∞	∞	B	AB	BC, BE, BF
2	d	0	4	1	3	8	10	∞	F	AB, BF	FC, FE, FG
3	d	0	4	12	3	11	10	9	C	AB, BF, FC	-
4	d	0	4	12	3	11	10	9	E	AB, BF, FC, EF	EB, ED
5	d	0	8	12	6	11	10	9	G	AB, BF, FC, EF, GF	GD
6	d	0	8	12	7	11	10	9	D	AB, BF, FC, EF, GF DG	-

8

(c) A *feedback edge* set of an undirected graph $G = (V, E)$ is a subset of edges $E' \subseteq E$ that intersects every cycle of the graph. Thus, removing the edges E' makes the graph

Using the modified algorithm in (b) to find the MaxST

$T(V, E_T)$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

complexity?

13

6. [15 points] GENERALIZED SHORTEST PATHS
There are many applications, where it makes sense to consider weight costs not only on

Design and analyze an efficient algorithm for computing the minimum cost path from the start vertex s to each vertex $u \in V$.

But this time, v is relaxed only if $u.d + f(u,v) + g(v) < v.d$.
Still, every time we just pop the element with minimal value of d.
d. repeat c, until the Q is empty.

Complexity?