CS 3329 final

Name____________________________

True or false 3 points each

1. (**TRUE**) If P != NP then no problem in NP can be solved in polynomial time.
2. (**FALSE**) DFS can be used to find single source shortest path in a graph faster than Dijkstra's algorithm, sometimes.
3. (**TRUE**) In searching a tree, DFS is guaranteed to return a positive (found) or negative (not found) result, whereas BFS is not.
4. (**TRUE**) Given any graph, to find the minimal number of colors to color this graph is unsolvable.
5. (**FALSE**) Any problem can be solved with greedy algorithm can also be solved with dynamic programming.
6. (**TRUE**) Bellman Ford algorithm can only be used on directed graphs.
7. (**TRUE**) Bellman Ford algorithm can be used to detect negative cycles in the graph
8. (**TRUE**) All P problems are also NP problems
9. (**TRUE**) Dijkstra's Algorithm does not allow negative edges

10(**FALSE**) We can find the longest path in a graph with dynamic programming, but not with greedy algorithm.

Short answer:

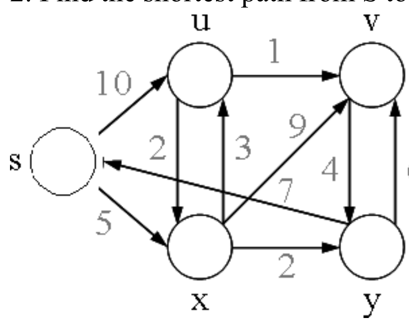
1. The following is the characters and their frequency in an article. Find an Huffman Code 5 points

character	Frequency
a	5
b	9
c	8
d	13
e	36
f	45
g	22

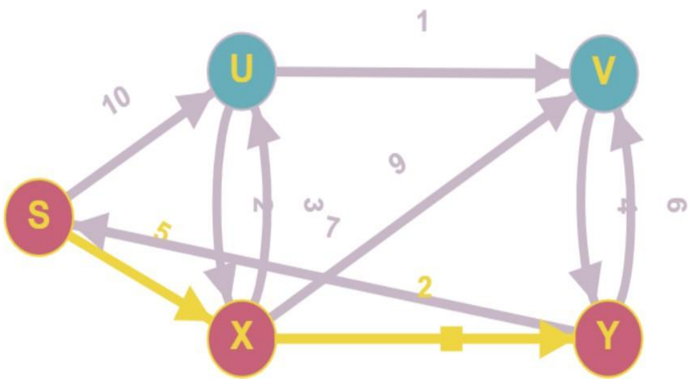
Answer:

Character	frequency	code
a	5	0100
b	9	0101
c	8	0110
d	13	0111
e	36	10
f	45	11
g	22	00

2. Find the shortest path from S to Y with Dijkstra's algorithm 8 points

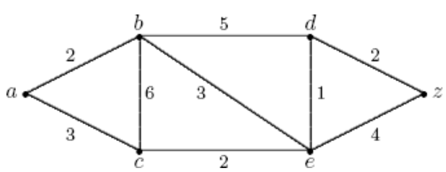


Answer:



The shortest path length is 2: S => X => Y

3. Use Prim's algorithm to find a minimum spanning tree in the following weighted graph. Use alphabetical order to break ties. 5 points



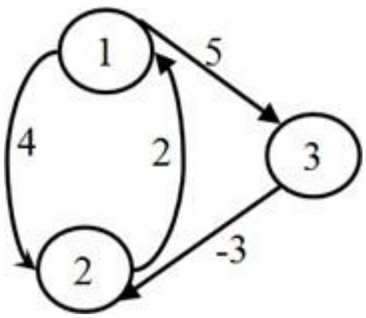
Answer:

S{a} – source
vertices = {b,c,d,e,f}

Step 1: taking lightest edge from source {a,b}
Step 2: after source, s{a,b}, lightest edge = s ={a.b}, {b,e}
Step 3: Now vertices = {c,d,f}, another edge = {e,d}
Step 4: Point c is left, taking light edge available from e {e,c}
Step 5: vertices = {f}. Taking light edge from d {d,f}

⇨ Total weight = {a,b} + {b,e} + {c,e} + {e,d} + {d,f}
= 2+3+2+1+2
= 10

4. For the following graph, run Floyd-Washall algorithm. 10 points



Answer:

-

4)

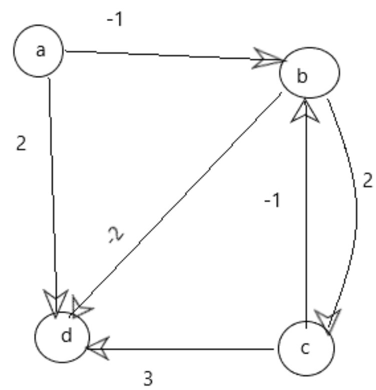
$$A^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 \\ 2 & 0 & \infty \\ \infty & -3 & 0 \end{bmatrix} \end{array}$$

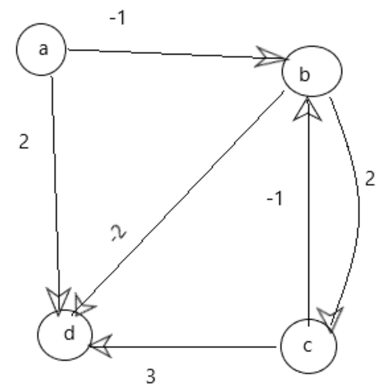
$$A^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 \\ 2 & 0 & \\ \infty & & 0 \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 \\ 2 & 0 & 7 \\ \infty & -3 & 0 \end{bmatrix} \end{array}$$

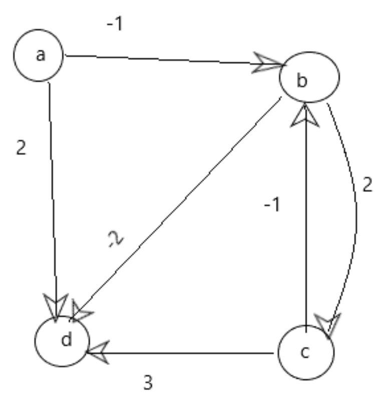
$$A^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & \\ 2 & 0 & 7 \\ & -3 & 0 \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 \\ 2 & 0 & 7 \\ -1 & -3 & 0 \end{bmatrix} \end{array}$$

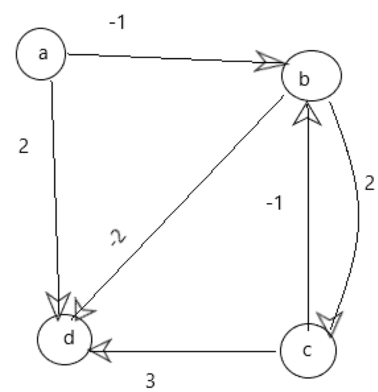
$$A^3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & & 5 \\ & 0 & 7 \\ -1 & -3 & 0 \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 2 & 5 \\ 2 & 0 & 7 \\ -1 & -3 & 0 \end{bmatrix} \end{array}$$

5. Bellman Ford Algorithm. Find shortest path from source a 8 points









Answer:

5)

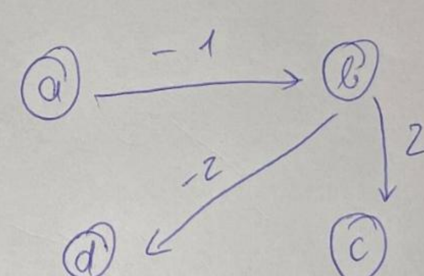
a	b	c	d
0	-1	1	-3

from a to:

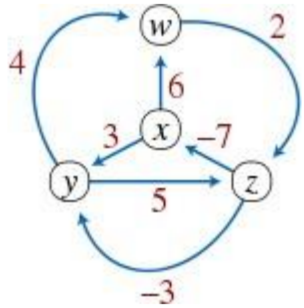
b = a → b

c = a → b → c

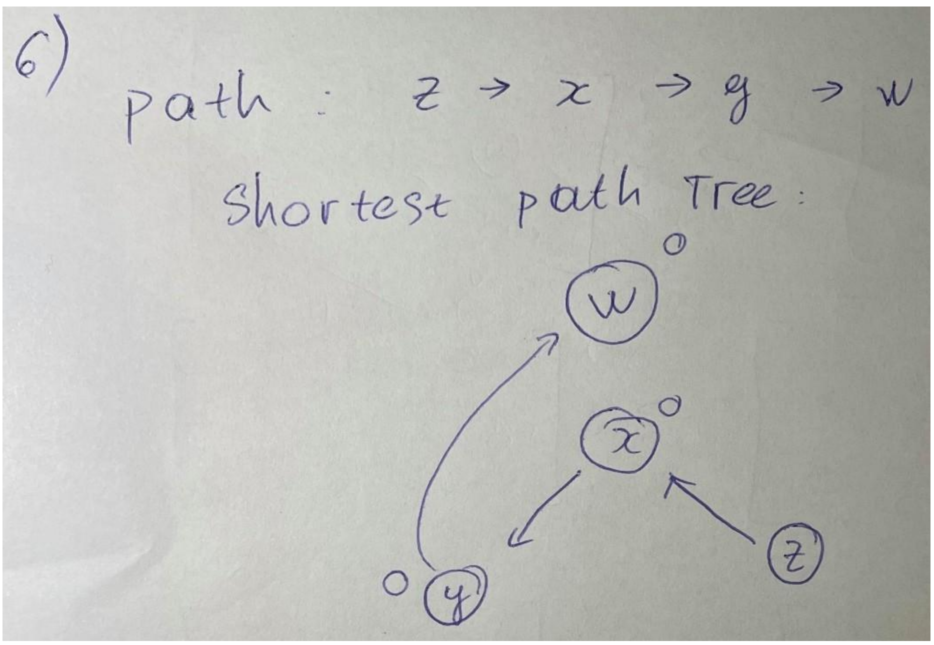
d = a → b → d



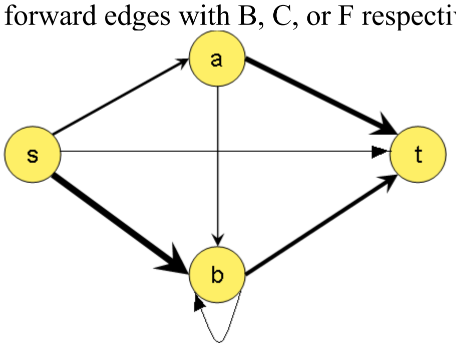
6. Use Johnson's algorithm to find the all source shortest paths in the following graph. 10 points



Answer:

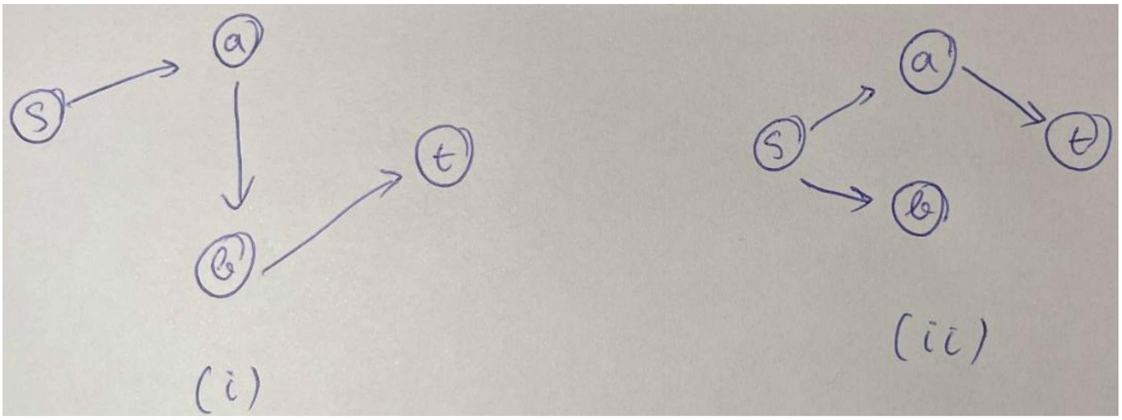


7. Run the DFS on the following graph, with s being the source. Identify the back, cross or forward edges with B, C, or F respectively.   8 points.



Answer:

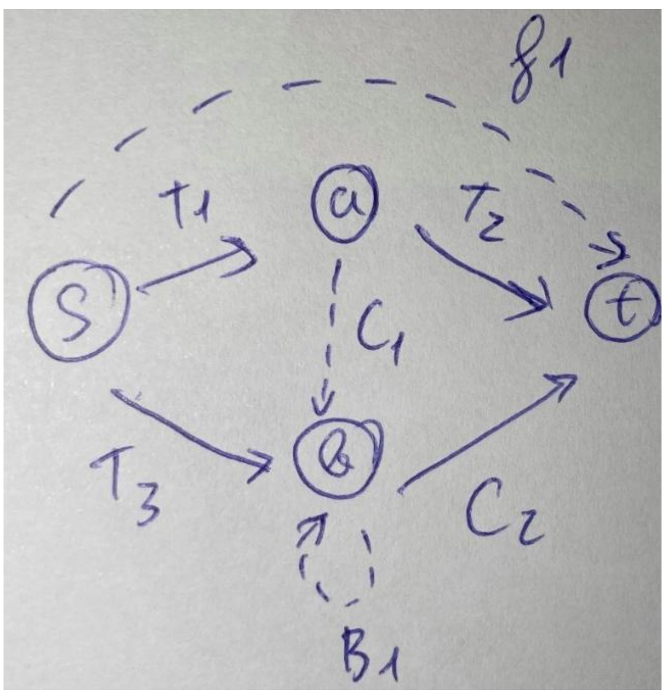
tree edge , backward edge and cross edge we made two DFT of given graph:



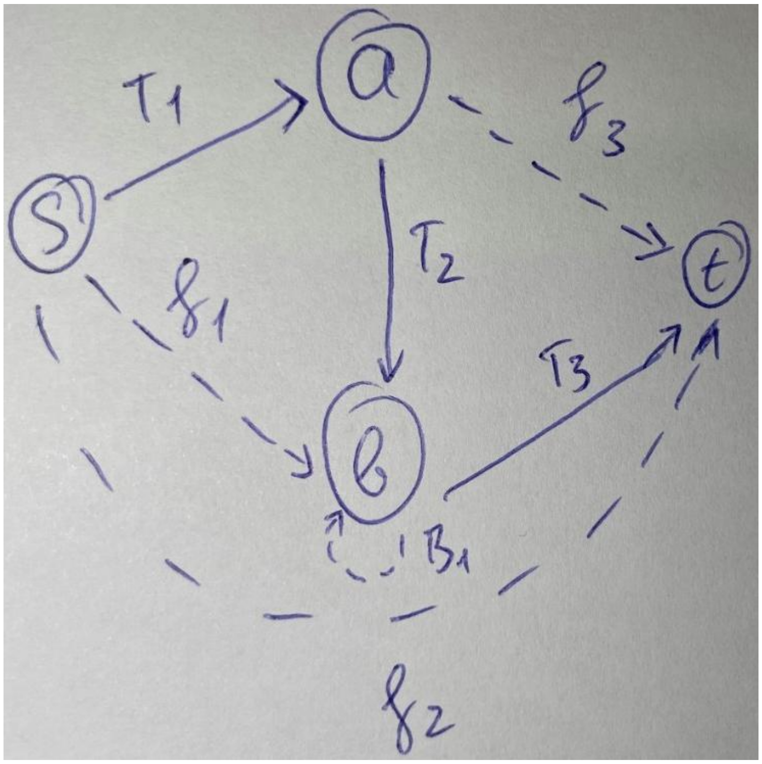
now, in below DFT t1,t2,t3 is showing tree edges.

c1,c2,c3 showing cross edges

f1,f2,f3 is showing forward edges



in above DFT dotted lines are showing tree edges back edges and cross edges.



8. Compute the prefix function for the pattern "ababacab" 8 points

Given Pattern : **"ababacab"**
We will consider below algorithm for step by step solution to compute prefix function,

Algorithm :

```
p ← "ababacab"
m ← length[p]
A[1] ← 0
k ← 0
for q ← 2 to m do
    while k > 0 and p[k+1] ≠ p[q] do
        k←A[k]
    end while
    if p[k+1] = p[q] then
        k ← k+1
    end if
    A[q] ← k
end for
```

Stepwise solution for computing prefix function of "ababacab" using above algorithm :
Initially **p ← "ababacab", m = length[p] = 8, A[1] = 0, k = 0**

Step 1)
q = 2, k = 0
A[2] = 0

q	1	2	3	4	5	6	7	8
p	a	b	a	b	a	c	a	b
A	0	0						

Step 2)
q = 3, k = 0
A[3] = 1

q	1	2	3	4	5	6	7	8
p	a	b	a	b	a	c	a	b
A	0	0	1					

Step 3)
q = 4, k = 1
A[4] = 2

q	1	2	3	4	5	6	7	8
p	a	b	a	b	a	c	a	b
A	0	0	1	2				

Step 4)
q = 5, k = 2
A[5] = 3

q	1	2	3	4	5	6	7	8
p	a	b	a	b	a	c	a	b
A	0	0	1	2	3			

Step 5)
q = 6, k = 3
A[6] = 0

q	1	2	3	4	5	6	7	8
p	a	b	a	b	a	c	a	b
A	0	0	1	2	3	0		

Step 6)
q = 7, k = 0
A[7] = 1

q	1	2	3	4	5	6	7	8
p	a	b	a	b	a	c	a	b
A	0	0	1	2	3	0	1	

Step 7)
q = 8, k = 1
A[8] = 2

q	1	2	3	4	5	6	7	8
p	a	b	a	b	a	c	a	b
A	0	0	1	2	3	0	1	2

A represents prefix function, **Answer : Prefix function A :**

0	0	1	2	3	0	1	2

9. Describe the steps in proving a problem is in NPC 8 points

The reducibility relation "$\leq_T$ " is transitive, i.e,
$A \leq_T B$ and $B \leq_T C$ imply $A \leq_T C$
Therefore, to prove that a problem A is NPC, we need to:
(1) show that $A \in NP$
(2) choose NPC problem B, i.e., $B \in NPC$,
classify a polynomial transformation T from B to A show that $B \leq_T A$