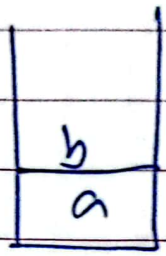
day / date:

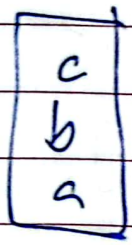
Assignment 4
21K-3153

Q1) a

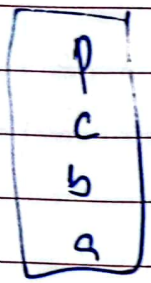
a, b:



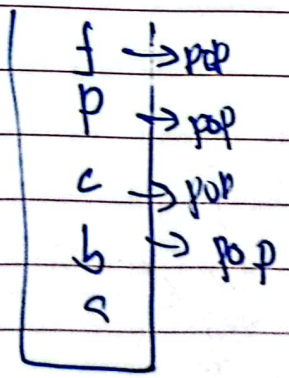
a, b, c:



a, b, c, p:



a b c p f:



after 1 pop:

a, b, c, p, f, d:

d
a

day / date:

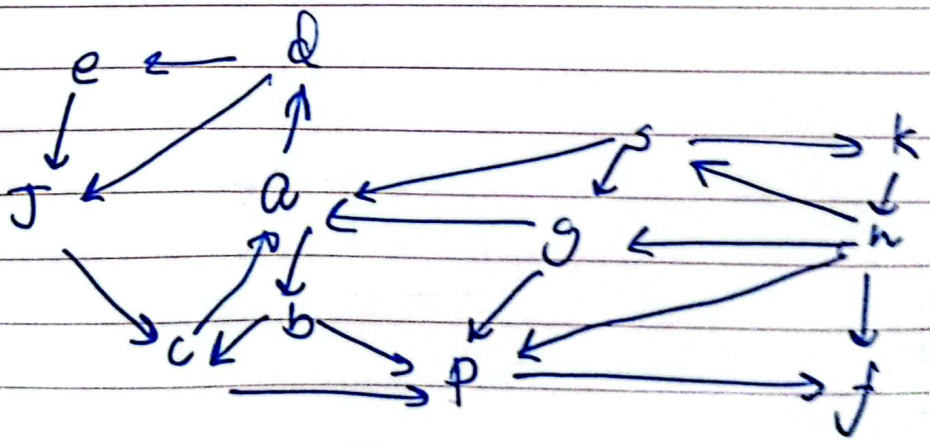
a,b,c,p,f,d,e : e d a

a,b,c,p,f,d,e,J : J e d a

DFS= a b c p f d e J

(b) back edges = {c, a} forward edges = {p, d}
cross edges = J, c

(c)

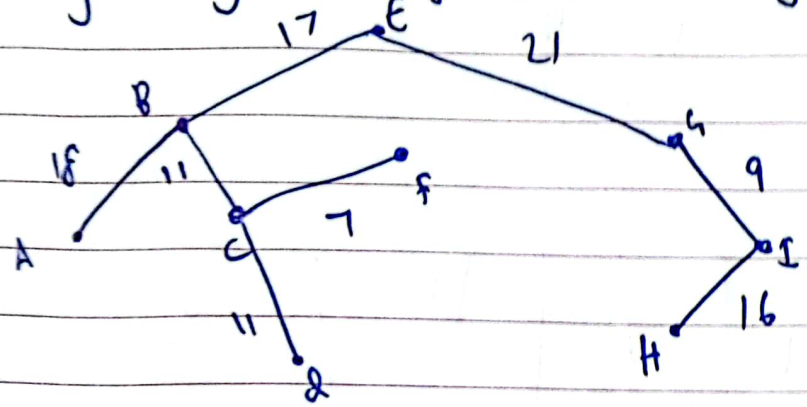


1 b f
2 g
3 p
4 h k
5 a b c d e J

day / date:

(a)

(b) Begins by selecting the lowest weighted edge



Cf

CI

BC

~~Bf~~ X

CD

1+1

BE

~~A~~B

~~A~~C X

~~E~~f X

~~D~~EG

4+17+21+9+16+7+11+11 = 110

(c) Loop runs for num of vertices (V) :-
O(V)
Checking edge : O(E log E)
Create graph : O(E log V)

(d) thus O(E log E) complexity.
If $E \leq V^2$ then O(E log V)

day / date:

(3) $D^0 = \begin{pmatrix} 0 & 8 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & \infty & 0 & 4 \\ \infty & 2 & 9 & 0 \end{pmatrix}$

$D^1 = \begin{pmatrix} 0 & 8 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 9 & 0 \end{pmatrix}$ | $D^2 = \begin{pmatrix} 0 & 8 & 9 & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 3 & 0 \end{pmatrix}$

$D^3 = \begin{pmatrix} 0 & 8 & 9 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 12 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{pmatrix}$ | $D^4 = \begin{pmatrix} 0 & 3 & 4 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 7 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{pmatrix}$

~~Matrix for~~ ~~for~~ n

traversing matrix = n^2

total iterations = total vertices

$n^2 \times n$

$O(n^3)$

(4) To prove Dijkstra, we need to verify our relaxation step is valid.

The invariant of the relaxation step is

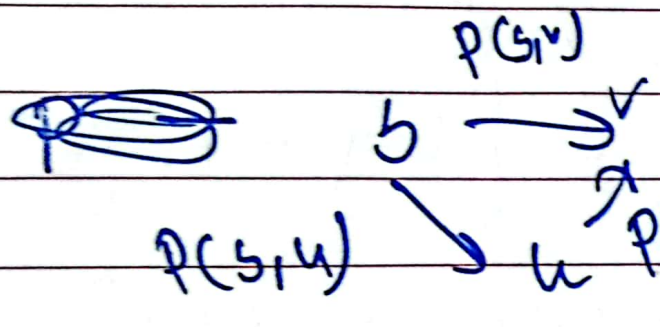
$$d[v] \geq p(s,v)$$

$p(s,v)$ = shortest path from $s-v$

$d[v]$ = length of current shortest path from $s-v$

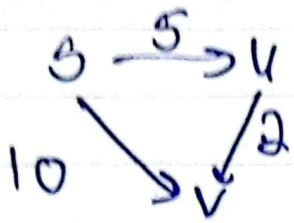
To prove, lets take the expression

$$p(s,v) \leq p(s,u) + p(u,v)$$



Shortest path of $s-v$ must be $\leq$ shortest path of $s-u$ + shortest of $u-v$

day / date:



Here, it is contradicted as $p(s,u) + p(u,v)$ is the shortest path. However, after relaxation, $p(s,v)$ become shortest as derived from $p(s,u)$ to $p(u,v)$

$$p(s,v) \leq p(s,u) + p(u,v)$$

$$p(s,u) = d[u] \quad p(u,v) = w(u,v)$$

$$p(s,v) \leq d[u] + w(u,v)$$

$$d[u] + w(u,v) = d[v]$$

$$p(s,v) \leq d[v]$$ ✓ as written previously

day / date:

(5)

dijkstra:

for each v in vertex:
    distance[v] ← ∞
    previous[v] ← null

if v not searched:
    add to priority queue Q
~~Distance[s]~~
Distance[current] = 0
while Q != null:
    max ← extract max from Q
~~for every neighbor:~~ for each vertex not searched for every neighbor of cur
    ~~dist ← distance[u]~~

    dist ← distance[current] + edge[u,v]
    if dist > distance[max]
        distance[max] ← dist
        previous[max] ← current
return distance

day / date:

(b)

s	b	c	d	e	f	g
	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
	12	5				
s, b	12	20	24	23	$-\infty$	$-\infty$
s, b, d	12	27	24	30	31	$-\infty$
s, b, d, f	12	40	24	46	31	41
s, b, d, f, e	12	40	24	46	31	59
c, b, d, f, e, g	12	40	24	46	31	57

40 c

(5)

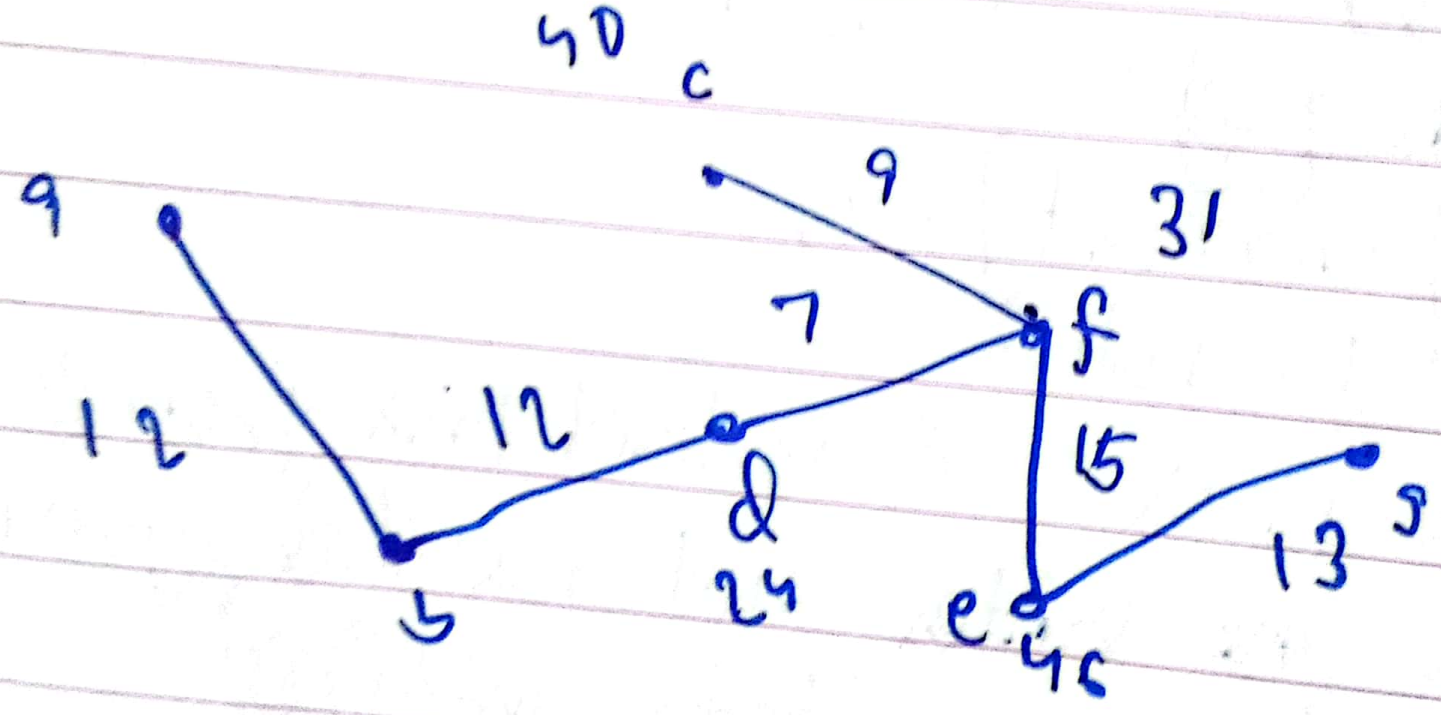
a, b = 12

a, b, d = 24

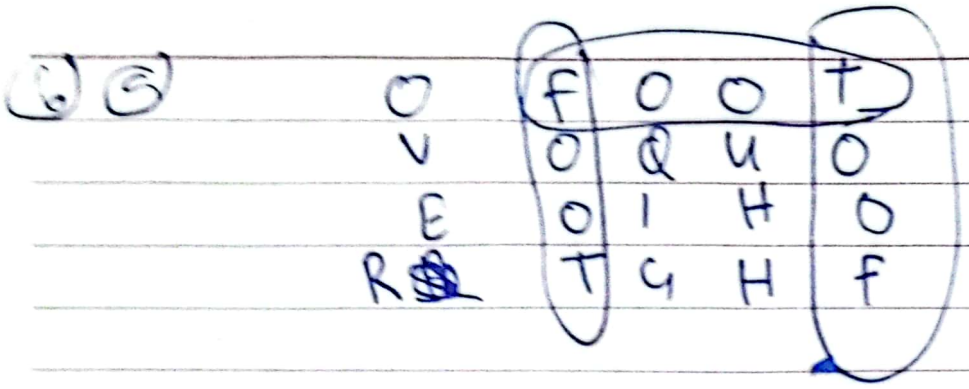
a, b, d, f, c = 31

a, b, d, f, e = 46

a, b, d, f, e, g = 59


40
c
9
9
31
7
f
12
12
15
d
24
13
9
e
46
59

day / date:



	F	O	O	T
	O	Q	U	O
	O	I	H	O
	T	G	H	F

(c) Start at 0,0 and traverse the matrix row first.
Traverse each element and see if it matches the first of the given word (i.e "F"). Store coordinates of first word.
As soon as the first word is matched, check the position of row, col and see if the the length is equal or less than the max/min of row and col (M or N)

only check for the next element in directions where the word can fit / M or N ≥ length of word
~~If multiple options~~ Examining M or N

If multiple options, choose row first.
and all directions looked

As soon as one element is matched, mark as done.
If whole word found, return positions of final letter and beginning letter.

Traverse matrix again for multiple occurences. Skips elements marked as "done"