Name: Farishta
ID: 19560
CS455: Algorithm

Q24- Use Bellman Ford's Algorithm to find the shortest path of a maze.

$|V| = n = 14$

$|V| - 1 = 14 - 1 = 13$

Relaxation

if $(d[u] + c(u,v \times d[v])$
$d[v] = d[u] + c(u,v)$

Step0: At the time of initialization, all the vertices except the source 'S' vertices are marked by ∞ and 'S' marked as 0

S	A	B	C	K	D	P	F	T	R	H	X	M	E
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Step 0 – 1st iteration
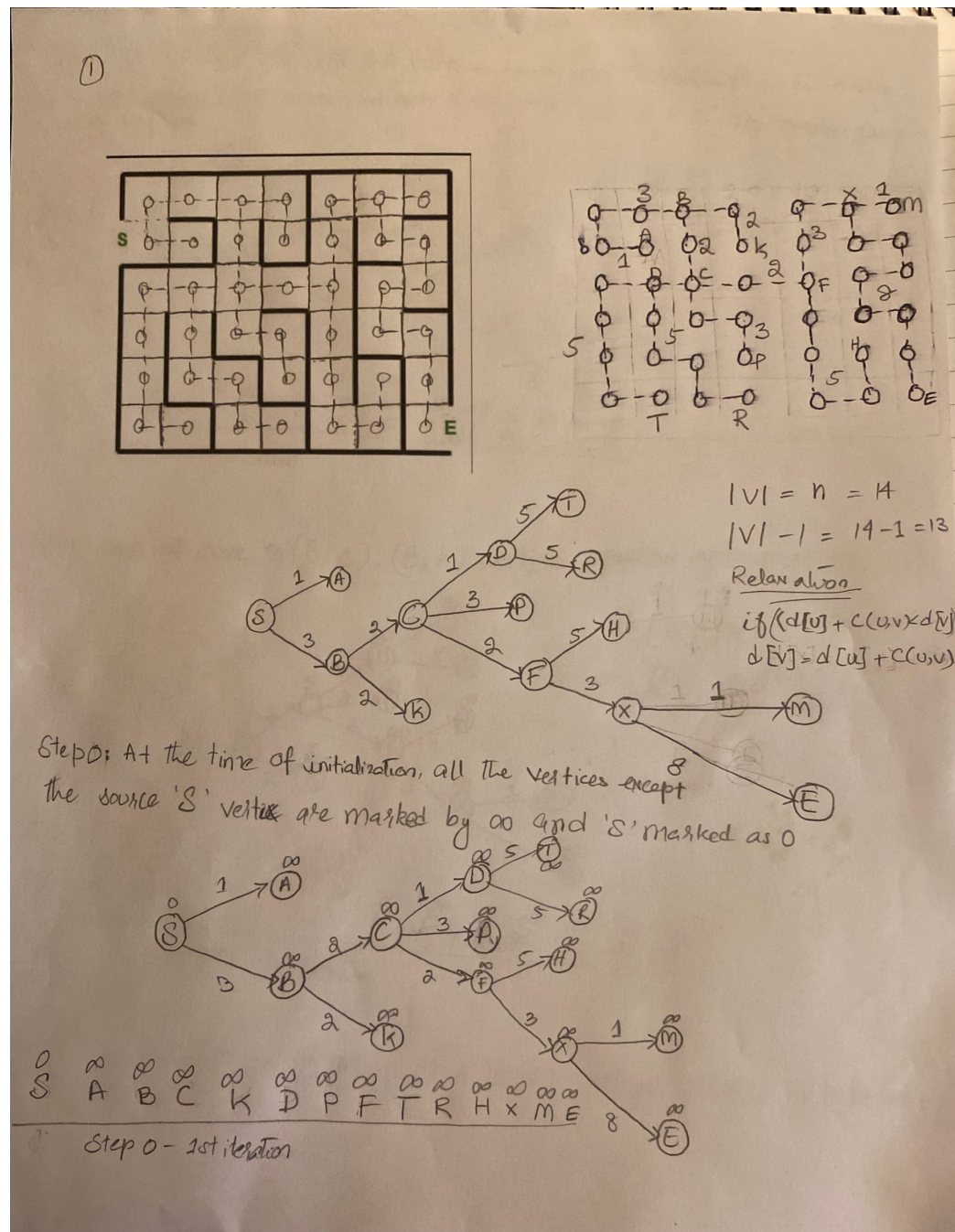
②

Step1: In the first step, all the vertices which are reachable from the Source, S, are updated by minimum cost. Hence vertices A & B are updated from ∞ to A=1 & B=3
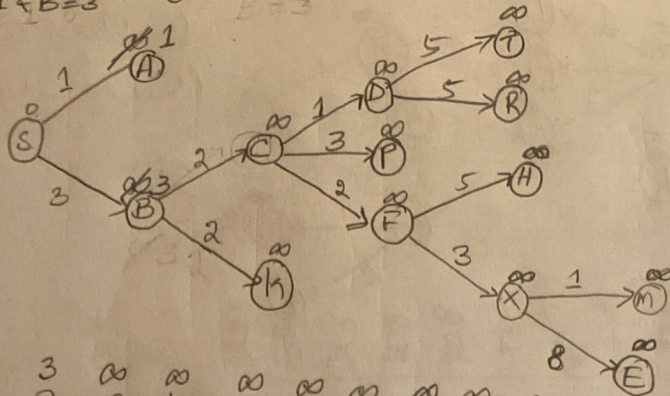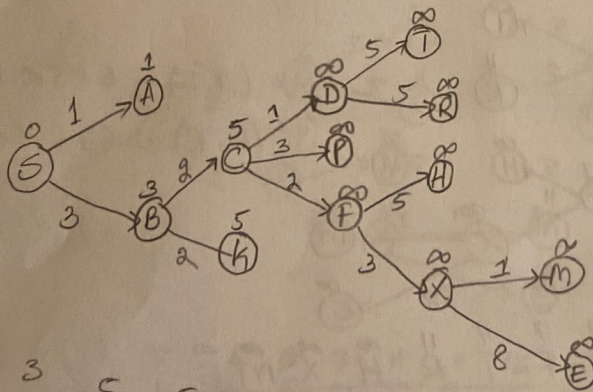

$(S, A) = 0 + 1 < ∞ = 1$

$(S, B) = 0 + 3 < ∞ = 3$

0	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
S	A	B	C	K	D	P	F	T	R	H	X	M	E

Step 1 - 1st iteration

Step2: Now will move to (B, C), (B, k) vertices, updated by minimum cost,

$(B, C) = 3 + 2 = 5 < ∞ = 5$

$(B, k) = 3 + 2 = 5 < ∞ = 5$

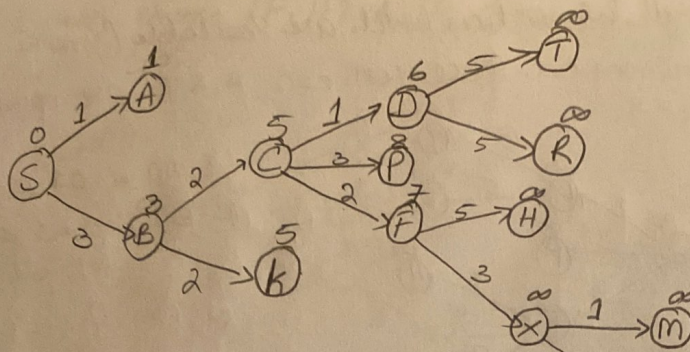| 0	1	3	5	5	∞	∞	∞	∞	∞	∞	∞	∞
S	A	B	C	K	D	P	F	T	R	X	M	E

Step2: 1st iteration

Step3: Now will pick (C, D), (C,P) & (C, f) vertices, updated by min cost,

③

Step3.

$(C,D) = 5+1 < \infty = 6$
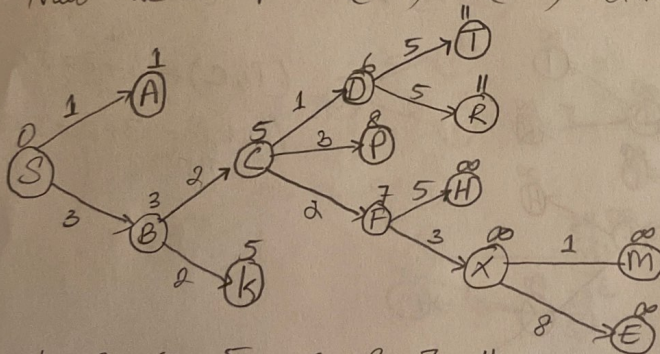$(C,P) = 5+3 < \infty = 8$
$(C,f) = 5+2 < \infty = 7$

0	1	3	5	5	6	8	7	∞	∞	∞	∞	∞
S	A	B	C	k	D	P	F	T	R	H	X	M	E

Steps: 1st iteration

Step4: Now we will pick (D,T) & (D,R) vertices to updated to the min cost
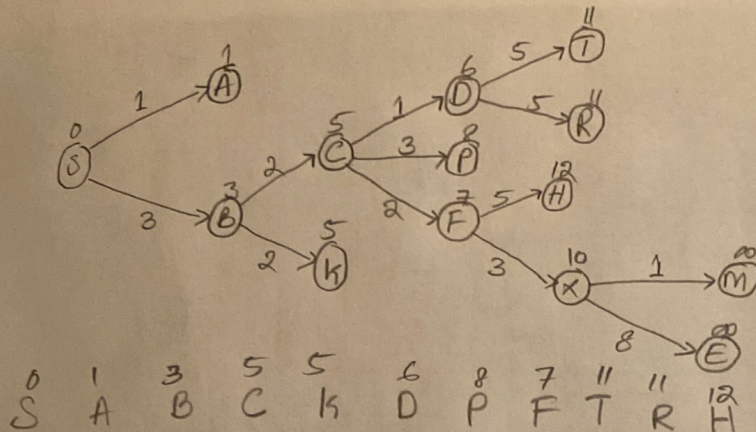

$(D,T) = 6+5 < \infty = 11$
$(D,R) = 6+5 < \infty = 11$

0	1	3	5	5	6	8	7	11	11	∞	∞	∞	∞
S	A	B	C	k	D	P	F	T	R	H	X	M	E

Step4: 1st iteration

Step5: Next will choose (F, H) & (F, X) vertices to updated to the min cost

(F, H) = 7 + 5 < ∞ = 12
(F, X) = 7 + 3 < ∞ = 10

0	1	3	5	5	6	8	7	11	11	12	10	∞	∞
S	A	B	C	k	D	P	F	T	R	H	X	M	E

Step5: 1st iteration

Step6: Finally Choose (X, M) & (X, E) vertices to updated to the min cost.
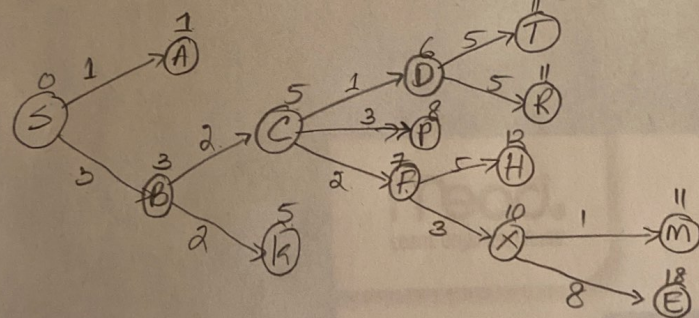
(X, M) = 10 + 1 < ∞ = 11
(X, E) = 10 + 8 X ∞ = 18

0	1	3	5	5	6	8	7	11	11	12	10	11	18
S	A	B	C	K	D	P	F	T	R	H	X	M	E

Step6: 1st iteration

Hence the minimum vertex b/w S & E is 18
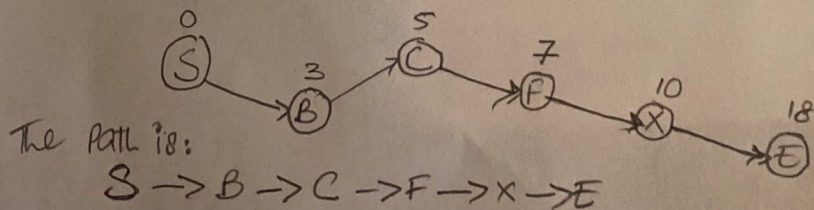
The Path is:

S → B → C → F → X → E

Dijkstra's algorithm is a Greedy algorithm and time complexity is O(VLogV) (with the use of Fibonacci heap). Dijkstra doesn't work for Graphs with negative weight edges, Bellman-Ford works for such graphs. Bellman-Ford is also simpler than Dijkstra and suites well for distributed systems. But time complexity of Bellman-Ford is O(VE), which is more than Dijkstra.