14.05.20

DAA Revision Test-2

Jovial Joe
Jayarson
IES17CS016

Q.) Explain Bellmann ford Algorithm.

Ans.

- Bellman ford algorithm is used to find the shortest distance between points 'u' & 'v' on a graph.
- Even though dijkstra's Algorithm does the same it will fail in a graph with negative weights.
- Steps in the algorithm:

Step 1: Select a vertex 'u' and mark it as having

$cost = 0 \Rightarrow d[u] = 0$

Step 2: In the same breath mark rest of the v vertices as ∞. $d[v_i] = \infty$

Step 3: To decide which is the cheapest adjacent vertex:

```
if ( d[u] + c(u, v) < d[u] ) {
        d[v] = d[u] + c(u, v)
}
```

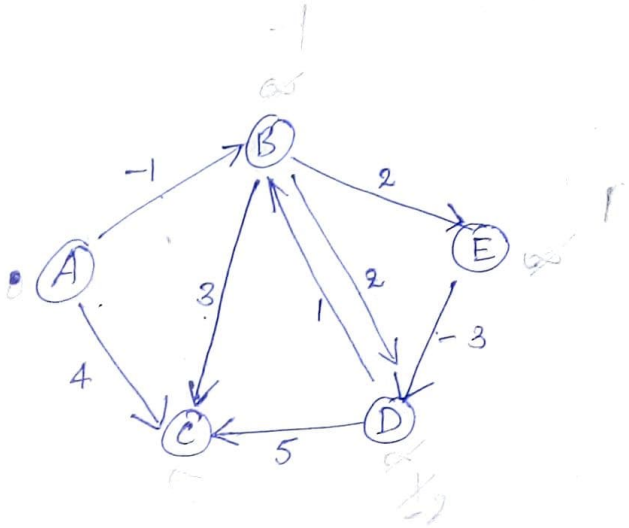
where d[u] or d[v] implies the current weight upon u & v.
And c(u, v) is the cost of travelling from u to v.

Step 4: Now if the graph contains n vertices repeat the above step for n-1 times.

Step 5: Stop if any iteration gives the same result.

eg:-

Given graph: G(E,V)



Pass:1

A	B	C	D	E
0	∞	∞	∞	∞

Pass 2:

A	B	C	D	E
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	2	∞	∞

Pass 3:

A	B	C	D	E
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	2	∞	∞
0	-1	2	∞	1
0	-1	2	1	1
0	-1	2	-2	1

Pass 4

A	B	C	D	E
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	2	∞	∞
0	-1	2	∞	1
0	-1	2	1	1
0	-1	2	-2	1

$V = \{A, B, C, D, E\}$

$E = \{\overrightarrow{AB}, \overrightarrow{AC}, \overrightarrow{BC}, \overrightarrow{BD}, \overrightarrow{BE}, \overrightarrow{DB}, \overrightarrow{DC}, \overrightarrow{ED}\}$

Eg. $d(A) = 0$ $\quad c(A,B,C) = -1+3 = 2$

$d(C) = 4$

$\Rightarrow d(A) + c(ABC) < d(C)$

$\Rightarrow d(C) = d(A) + c(A,B,C)$

$= 2$

0 -1 2 -2 1 ← repitition hence stop iteration.

Since in a graph there can be n-vertices and n-edges then total/max no of iteration (including sub) = m×n

⇒ Complexity = O(m×n) → polynomial time.