5. Bellman-Ford's Algorithm

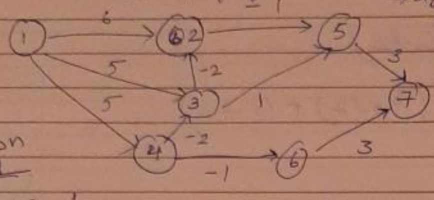
28 February 2021 08:40

SSSP → k → number of ~~no~~ edges

given source, find shortest path from source to all the vertices

- Djikstra's ↳ minimal → greedy
- Bellman-Ford's ↳ dynamic

Bellman Ford



$dist^k[u] = \min \{ dist^{k-1}[u], \min_i \{ dist^{k-1}[i] + cost[i, u] \} \}$

Source : 1

number of edges → k	1	2	3	4	5	6	7
1	0	6	5	5	α	α	α
2	0	3	3	5	5	4	α
3	0	1	3	5	~~α~~ 2	~~α~~ 4	~~α~~ 7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3
7	0	1	3	5	0	4	3

5

$d^1(6) = \min\{\alpha, 5 + (-1)\}$
$= 4$

$d^3(2) = \min\{3, 5-2-2\}$

1-4-3-2

$d^3(5)$ min $\{5, 4$

min $\{2, 4\}$ (2) (3)

$d^3(6) = \min\{4, \alpha\}$

$d^4(2) = \min\{1,$

$d^3(7) = \min\{\alpha, 5+3, \}$

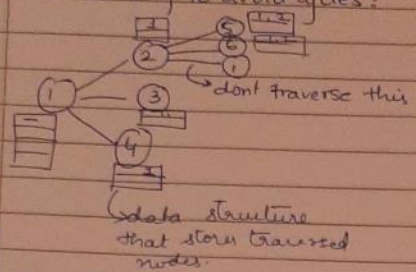
$d^4(5) = \min\{2, 1-1, 4+3, 3+1\}$

Complexity → $O(n^3)$

To avoid cycles, avoid checking for more than n-1 edges for a graph with n edges

```
BellmanFord(v, cost, dist, n)
{ for i=1 to n do
      dist(i) = cos[v,i]
  for k=2 to n-1
      for each u such that u≠v and u has atleast 1 incoming edge do
          for each <i,u> in the graph
              if dist[u] > dist[i] + cost[i,u]
                  dist[u] = dist[i] + cost[i,u]
```

Another way to avoid cycles:



* Space complexity greater if cycles are avoided

TEST 1 PORTION

~~Dijkstra~~

1. divide & conquer
2. greedy
3. dynamic
4. backtracking
5. branch and bound

(backtracking, branch and bound) → exponential 2^n, n^n or $n!$