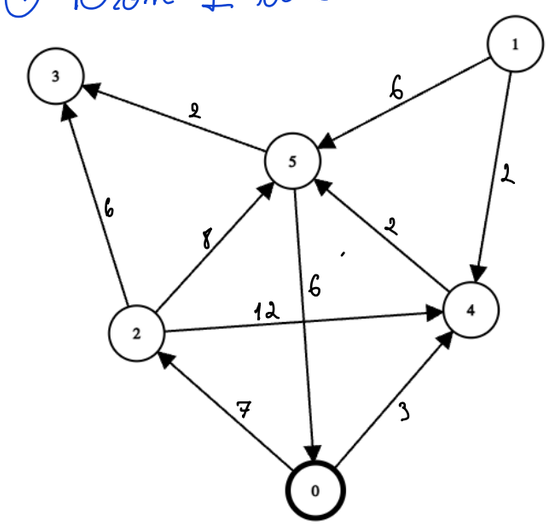
Assignment 3

Wednesday, 26 April 2023 13:12

Lowest cost path between s and t using Dijkstra's Algorithm

① From 1 to 3



s = 1, t = 3	x	y	cost: dictionary	Q: priority queue	prev: dictionary
initialization			0: 0, 1: ∞, 2: ∞, 3: ∞, 4: ∞, 5: ∞	← (0,1) ←	0: ∅, 1: -1, 2: ∅, 3: ∅, 4: ∅, 5: ∅
iteration 1	1	4, 5	0: 0, 1: ∞, 2: ∞, 3: ∞, 4: 2, 5: 6	← (2,4) (6,5) ←	0: ∅, 1: -1, 2: ∅, 3: ∅, 4: 1, 5: ∅
iteration 2	4	5	0: 0, 1: ∞, 2: ∞, 3: ∞, 4: 2, 5: 4	← (4,5) (6,5) ←	0: ∅, 1: -1, 2: ∅, 3: ∅, 4: 1, 5: 4
iteration 3	5	0, 3	0: 0, 1: ∞, 2: ∞, 3: 6, 4: 2, 5: 4	← (6,3) (6,5) (100,0) ←	0: ∅, 1: -1, 2: ∅, 3: 5, 4: 1, 5: 4
iteration 4	3		0: 0, 1: ∞, 2: ∞, 3: 6, 4: 2, 5: 4	← (6,5) (100,0) ←	0: ∅, 1: -1, 2: ∅, 3: 5, 4: 1, 5: 4

relaxation ←

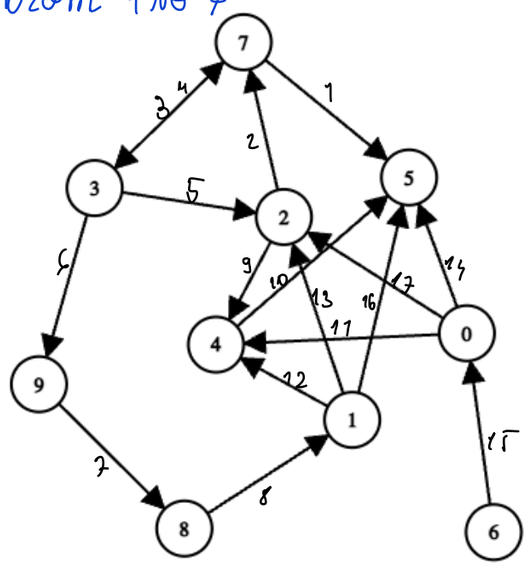
STOP ←

The minimum cost walk from s = 1 to t = 3 is built backwards from prev

t = 3, prev[3] = 5, prev[5] = 4, prev[4] = 1 = s

1 → 4 → 5 → 3, cost[3] = 6

② From 4 to 7



s = 4, t = 7	x	y	cost: dictionary	Q: priority queue	prev: dictionary
initialization			0: ∞, 1: ∞, 2: ∞, 3: ∞, 4: 0, 5: ∞, 6: ∞, 7: ∞, 8: ∞, 9: ∞	← (0,4) ←	0: ∅, 1: ∅, 2: ∅, 3: ∅, 4: -1, 5: ∅, 6: ∅, 7: ∅, 8: ∅, 9: ∅
iteration 1	4	5	0: ∞, 1: ∞, 2: ∞, 3: ∞, 4: 0, 5: 10, 6: ∞, 7: ∞, 8: ∞, 9: ∞	← (10,5) ←	0: ∅, 1: ∅, 2: ∅, 3: ∅, 4: -1, 5: 4, 6: ∅, 7: ∅, 8: ∅, 9: ∅
iteration 2	5		0: ∞, 1: ∞, 2: ∞, 3: ∞, 4: 0, 5: 10, 6: ∞, 7: ∞, 8: ∞, 9: ∞	←	0: ∅, 1: ∅, 2: ∅, 3: ∅, 4: -1, 5: 4, 6: ∅, 7: ∅, 8: ∅, 9: ∅

relaxation ←

Since the queue is empty, we can't compute the cost from 4 to 7, because there is no path