Lab 3: Lowest cost walk between s and t using Dijkstra's algorithm

S=5 t=1	x	y	queue	visited	dist-dictionary	Prev-dictionary
init			<| 5 |<	{5}	1|2|3|4|5 | | | |0	1|2|3|4|5 | | | |
It 1 it 1.1 it 1.2	5	 3 4	 <| 3 |< <| 4 |<	 {5,3} {5,3,4}	 1|2|3|4|5 | |5|3|0	 1|2|3|4|5 | |5|5|0
It 2 it 2.1	3	 1	 <| 1 |<	 {5,3,4,1}	 1|2|3|4|5 6| |5|3|0	 1|2|3|4|5 3| |5|5|0
It 3 it 3.1	4	 2 3	 <| 2 |< <| 3 |<	 {5,3,4,1,2} {5,3,4,1,2}	 1|2|3|4|5 5|10|4|3|0	 1|2|3|4|5 3|4|4|5|0
It 4 it 4.1	2	 1		 {5,3,4,1,2}	 1|2|3|4|5 5|10|4|3|0	 1|2|3|4|5 3|4|4|5|0

The path is built from prev, being { 5 > 4 > 3 >1 }, length = 6.

S=1 t=5	x	y	queue	visited	dist-dictionary	Prev-dictionary
init			<| 1 |<	{1}		1|2|3|4|5 | | | |
It 1 it 1.1 it 1.2	1	 3 2	 < | 3 |< <| 2 |<	 {1,3} {1,3,2}	 1|2|3|4|5 |6|2| |0	 1|2|3|4|5 |1|1| |
It 2 it 2.1	3	 2			 1|2|3|4|5 |6|2| |	 1|2|3|4|5 | 1 | 1 | |

5 has no parent in the prev dictionary therefore it cannot be reached from 1.