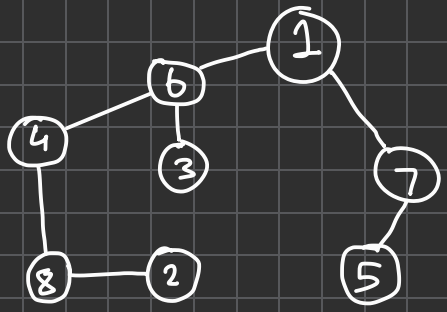
1)

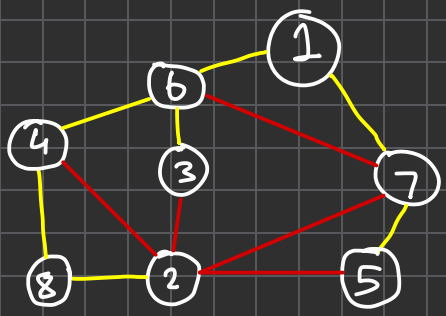
2)

i)



Cost: 19

ii)



{2,8} {4,8} {1,7} {3,6}
1 1 2 2

~~{2,4}~~ {1,6} {5,7} {4,6}
~~3~~ 4 4 5

~~{2,5}~~ ~~{2,3}~~ ~~{1,7}~~ ~~{2,7}~~
~~6~~ ~~7~~ ~~8~~ ~~12~~

Order: {2,8} {4,8} {1,7} {3,6} {1,6} {5,7} {4,6}

iii)

iteration	Vertices	Cost @ V: {1,2,3,4,5,6,7,8}
0	{1,2,3,4,5,6,7,8}	{0,∞,∞,∞,∞,∞,∞,∞}
1	{2,3,4,5,6,7,8}	{0,∞,∞,∞,∞,4,2,∞}
2	{2,3,4,5,6,8}	{0,12,∞,∞,4,4,2,∞}
3	{2,3,4,6,8}	{0,6,∞,∞,4,4,2,∞}
4	{2,3,4,8}	{0,6,2,5,4,4,2,∞}
5	{2,4,8}	{0,6,2,5,4,4,2,∞}
6	{2,8}	{0,3,2,5,4,4,2,1}
7	{2}	{0,1,2,5,4,4,2,1}
8	{ }	{0,1,2,5,4,4,2,1}

3)

Algorithm

1) Find cycle caused by new edge using DFS [O(N)]

2) Find edge with highest weight in cycle [O(N)]

3) Remove this edge [O(1)]

∴ Total runtime O(N)

4)

i) order = Sort (Service_time, increasing)

ii) Suppose $k = j+1$, $t_{L(j)} = 5$, $t_{L(k)} = 6$

Let $T(j-1) = x$

Current order: $x + (x+5) + 2x+5+x+6 = 5x+16$

New order: $x + (x+6) + (2x+6) + x + 5 = 5x+17$

iii) $O(n \log n)$ for sorting