Prim Algorithm: Minimum Spanning Tree

Data structures

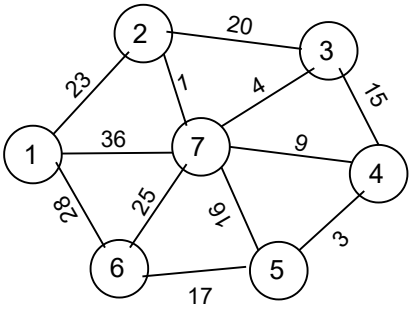
1. U set and V-U set
1.1 U set: vertexes which become MST members
1.2 V-S set: vertexes which are not MST members yet
2. lowcost[] least weights of each vertex
3. closest[] vertexes in V-S which closest to U

Always in V-U set

1. update lowcost[] and closest[] using current adjacent edges
2. find the index of least in lowcost[]

Conditions of update

```
if lowcost[i] > weight[curr][i] {
    lowcost[i] = weight[curr][i]
    closest[i] = curr
}
```



1	->	2(23) 6(28) 7(36)
2	->	1(23) 3(20) 7(1)
3	->	2(20) 4(15) 7(4)
4	->	3(15) 5(3) 7(9)
5	->	4(3) 6(17) 7(16)
6	->	1(28) 5(17) 7(25)
7	->	1(36) 2(1) 3(4) 4(9) 5(16) 6(25)

Initialization S={}, V-S={1,2,3,4,5,6,7}

X	1	2	3	4	5	6	7
lowcost	0	∞	∞	∞	∞	∞	∞
closest	-1	-1	-1	-1	-1	-1	-1

Step1 S={1}, V-S={2,3,4,5,6,7}

X	1	2	3	4	5	6	7
lowcost	0	23	∞	∞	∞	28	36
closest	-1	1	-1	-1	-1	1	1

Step2 S={1,2}, V-S={3,4,5,6,7}

X	1	2	3	4	5	6	7
lowcost	0	23	20	∞	∞	28	1
closest	-1	1	2	-1	-1	1	2

Step3 S={1,2,7}, V-S={3,4,5,6}

X	1	2	3	4	5	6	7
lowcost	0	23	4	9	16	25	1
closest	-1	1	7	7	7	7	2

Step4 S={1,2,7,3}, V-S={4,5,6}

X	1	2	3	4	5	6	7
lowcost	0	23	4	9	16	25	1
closest	-1	1	7	7	7	7	2

Step5 S={1,2,7,3,4}, V-S={5,6}

X	1	2	3	4	5	6	7
lowcost	0	23	4	9	3	25	1
closest	-1	1	7	7	4	7	2

Step6 S={1,2,7,3,4,5}, V-S={6}

X	1	2	3	4	5	6	7
lowcost	0	23	4	9	3	17	1
closest	-1	1	7	7	4	5	2

Step7 S={1,2,7,3,4,5,6}, V-S={}

X	1	2	3	4	5	6	7
lowcost	0	23	4	9	3	17	1
closest	-1	1	7	7	4	5	2