Greedy Method

Introduction

- Problem can be solved by a sequence of decisions. The greedy method has that each decision is locally optimal. These locally optimal solutions will finally add up to a globally optimal solution.
- approaches:

✓ Feasible

✓ Locally optimal

✓ Irrevocable

MST

- A Minimum Spanning Tree (MST) is a subgraph of an undirected graph such that the subgraph spans (includes) all nodes, is connected, is acyclic, and has minimum total edge weight
- Methods:

✓ Prim’s Algorithm

✓ Kruskal’s Algorithm

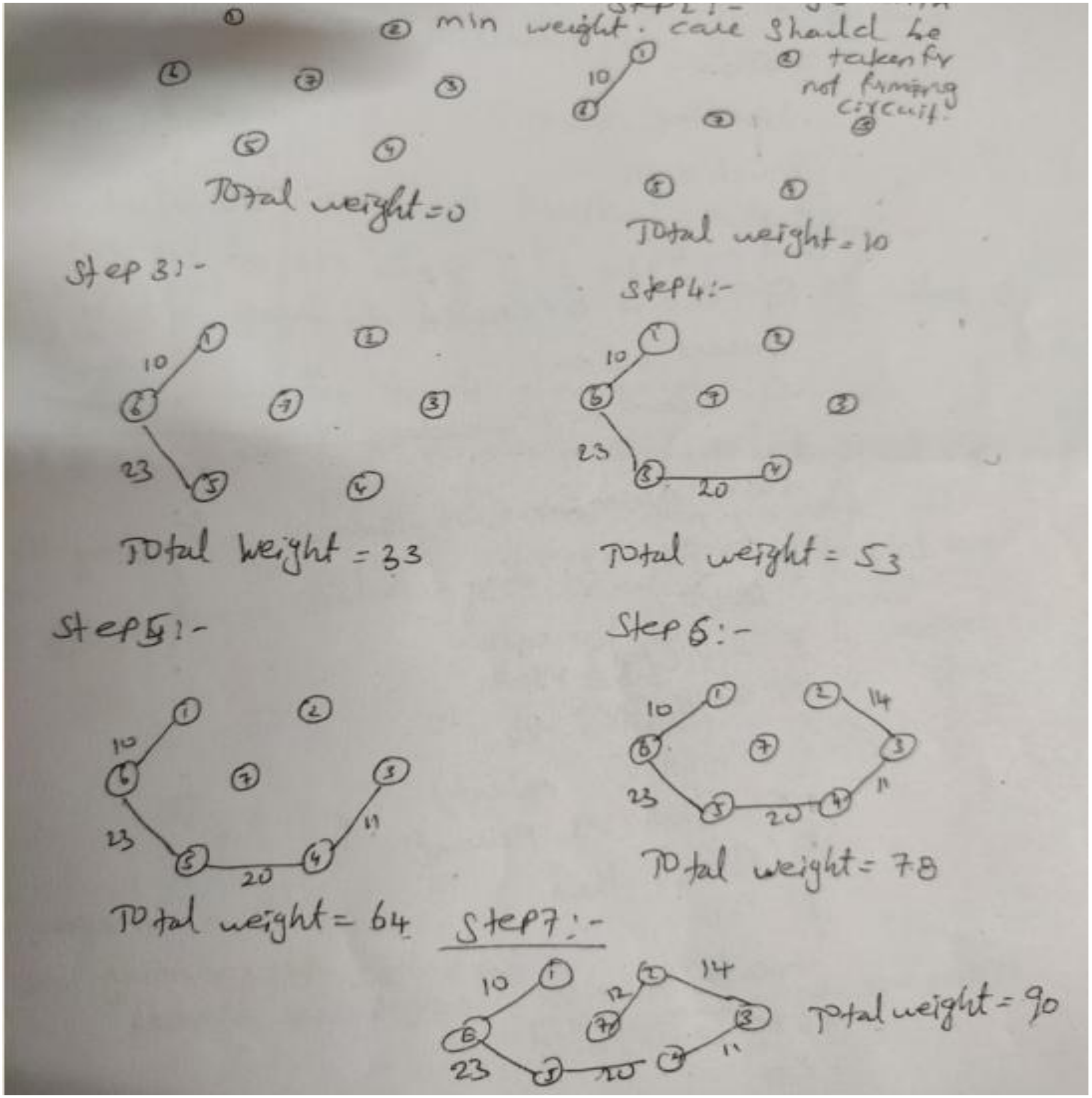
Prim’s Algorithm

- Consider a graph give below now, we will consider all the vertices first.
- Then we will select an edge with minimum weight.
- The algorithm proceeds by selecting adjacent edges with min weight.
- Care should be taken for not forming circuit.
- Analysis: $\Theta(v^2)$

Algorithm

- Input: A connected weighted graph with vertices V and edges E.
- Output: V_{new} and E_{new} describe a minimal spanning tree

for i← 0 to nodes-1 do
tree[i] ← 0
tree[0] ← 1
for k ← 1 to nodes do
  min dist ← ∞
for i ← 0 to nodes -1
  for j ← 0 to nodes -1
  if (G[i,j] AND ((tree[i] AND tree[j]) OR (tree[i] AND tree[j]))) then
if (G[i,j] < min dist) then
  min dist ← G[ij]
    v1 ← i
   v2← j
Write(v1,v2,min dist)
Tree[v1] ← tree[v2] ←1
Total ← total + min dist


min weight. care should be taken for not forming circuit.
Total weight = 0
Total weight = 10
Step 3:-
Total weight = 33
Step 4:-
Total weight = 53
Step 5:-
Total weight = 64
Step 6:-
Total weight = 78
Step 7:-
Total weight = 90

Kruskal's algorithm

- Consider a graph given below 1st we will select all the vertices.
- Then an edge with optimum weight is selected from heap, even though it is not adjacent to previously selected edge.
- care should be taken for not forming circuit.
- Analysis : $\Theta(E \log E)$

Algorithm

$Et \leftarrow \phi$; encounter $\leftarrow 0$

$K \leftarrow 0$

While encounter < $|v|-1$ do

$K \leftarrow k+1$

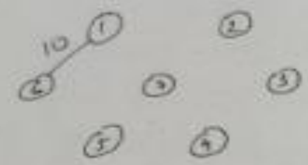
If Et u{eik} is acyclic

$Et \leftarrow Et \cup \{eik\}$

Encounter $\leftarrow$ encounter+1

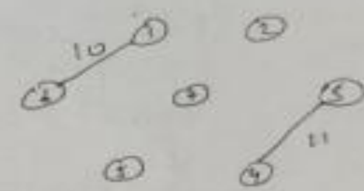
return Et

Step 2:-



Total weight =10

Step 3:-



Total weight = 21

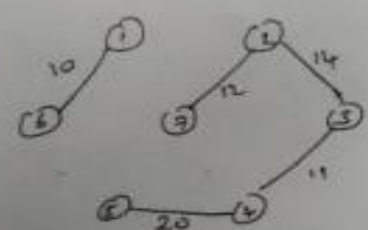
Step 4:-

Total weight = 33

Step 5:-

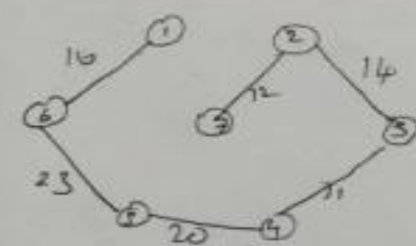
Total weight = 47

Step 6:-



Total weight = 67

Step 7:-



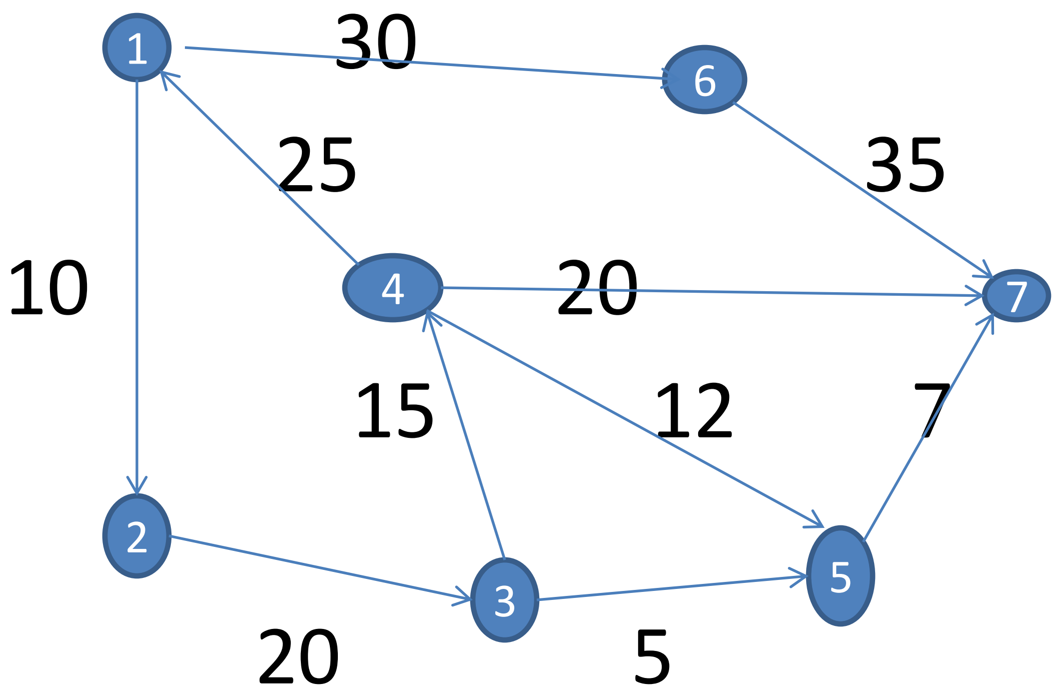
Total weight = 90

DIJIKSTRA's Algorithm

- Single source shortest path problem the shortest distance from a single vertex called source is obtained.
- Let G(V,E) be graph, find shortest path from vertex V_0 to all remaining vertex.
- Vertex V_0 is called source and last vertex is called destination.
- Time complexity O(n2)

Algorithm

```
Single_short_path(p,cost,dist,n)
{
For i←1ton do
{
S[i]←0;
Dist←cost[p,i];
}
S[p]←1
Dist [p]←0;
For val←2 to n-2 do
{
Dist[q]=min{dist[i]};
S[q]←1
For(all nodes r adjacent to q with s[r]=o)do
If(dist[r]>(dist[q]+cost[p,q]))then
Dist[r]←dist[q]+dist[p+q];
}
}
```


1
30
6
25
35
10
4
20
7
15
12
7
2
3
5
20
5

Vertex 1

$\{1,2\}=10$

$\{1,3\}=\infty$

$\{1,4\}= \infty$

$\{1,5\}= \infty$

$\{1,6\}=30$

$\{1,7\}= \infty$

Min dist is $\{1,2\}=10$

Vertex 3

{1,2,3}=30
{1,2,4}= ∞
{1,2,7}= ∞
{1,2,5}= ∞
{1,2,6}= ∞
Min dist {1,2,3}=30

Vertex 5

{1,2,3,4}=45
{1,2,3,5}=35
{1,2,3,6}= ∞
{1,2,3,7}= ∞
Min dist {1,2,3,5}=35

Vertex 7

{1,2,3,5,6}= ∞

{1,2,3,5,7}=42

Min dist {1,2,3,5,7}=42

Path	Weight
1,2	10
1,2,3	30
1,2,3,4	45
1,2,3,5	35
1,2,3,4,7	45
1,2,3,5,7	42
1,6	30
1,6,7	65

Huffman algorithm

- Generates high frequency with min no of bits.
- Generates low frequency with max no of bits.
- It is bottom up approach.
- Steps to be followed:

✓ Arrange the given message sources in descending order.

✓ Add last 2 source symbol into single unit & consider it as new source.

✓ Arrange source in descending order. Consider new message source obtained in step 2.

✓ Continue process until only 2 new source messages are left.

✓ Start assigning codes(0,1) in backward direction towards initial stage.