Constructing the Graph using its Cycle Space and finding its branches uniquely

Group 10

Introduction

Spanning Subgraph

A spanning subgraph of G(V,E) is a graph g(V,E’) where |E’| <= |E| and E’ can contain any number of edges, even zero edges.

Eulerian Subgraph

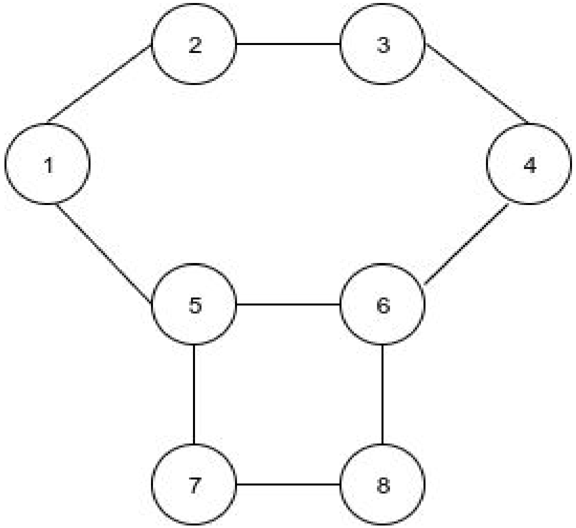
An eulerian subgraph of G(V,E) is any subgraph g(V,E’) where |V’| <= |V| and |E’| <= |E| such that every vertex in V’ has even degree.

Cycle Space

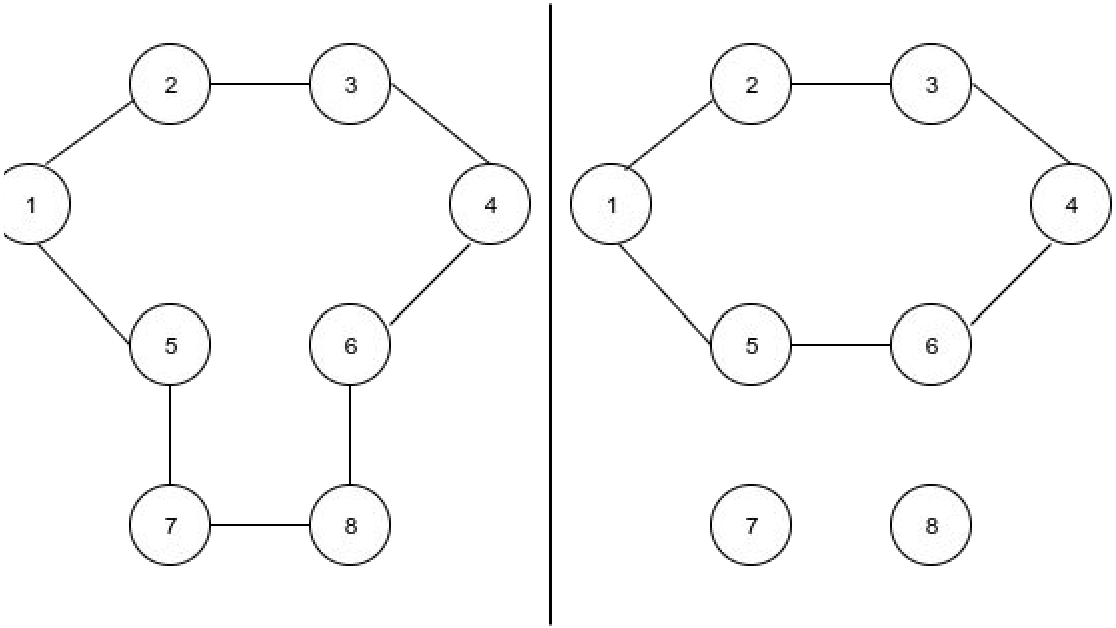
It is the collection of all the eulerian spanning subgraphs in the Graph.

Branch

A branch is simply any edge present in the spanning tree of a graph.



Given graph (G(V,E))



Two eulerian subgraphs for G(V,E)

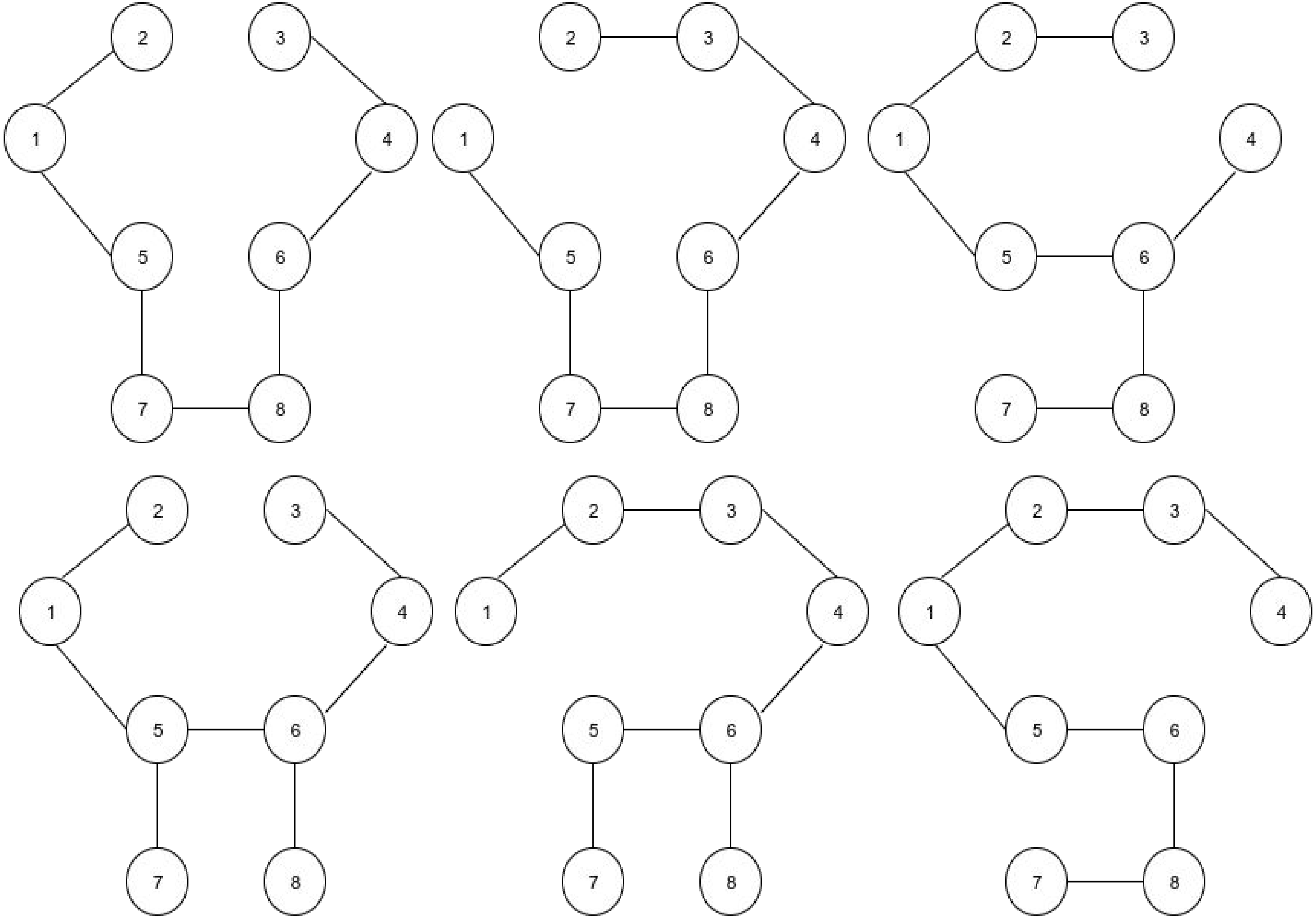


Fig. Some spanning trees for G(V,E)

Approach { $O(Log(V)*2^{E})$ }

For Graph Construction

Algorithm 1 Graph Construction

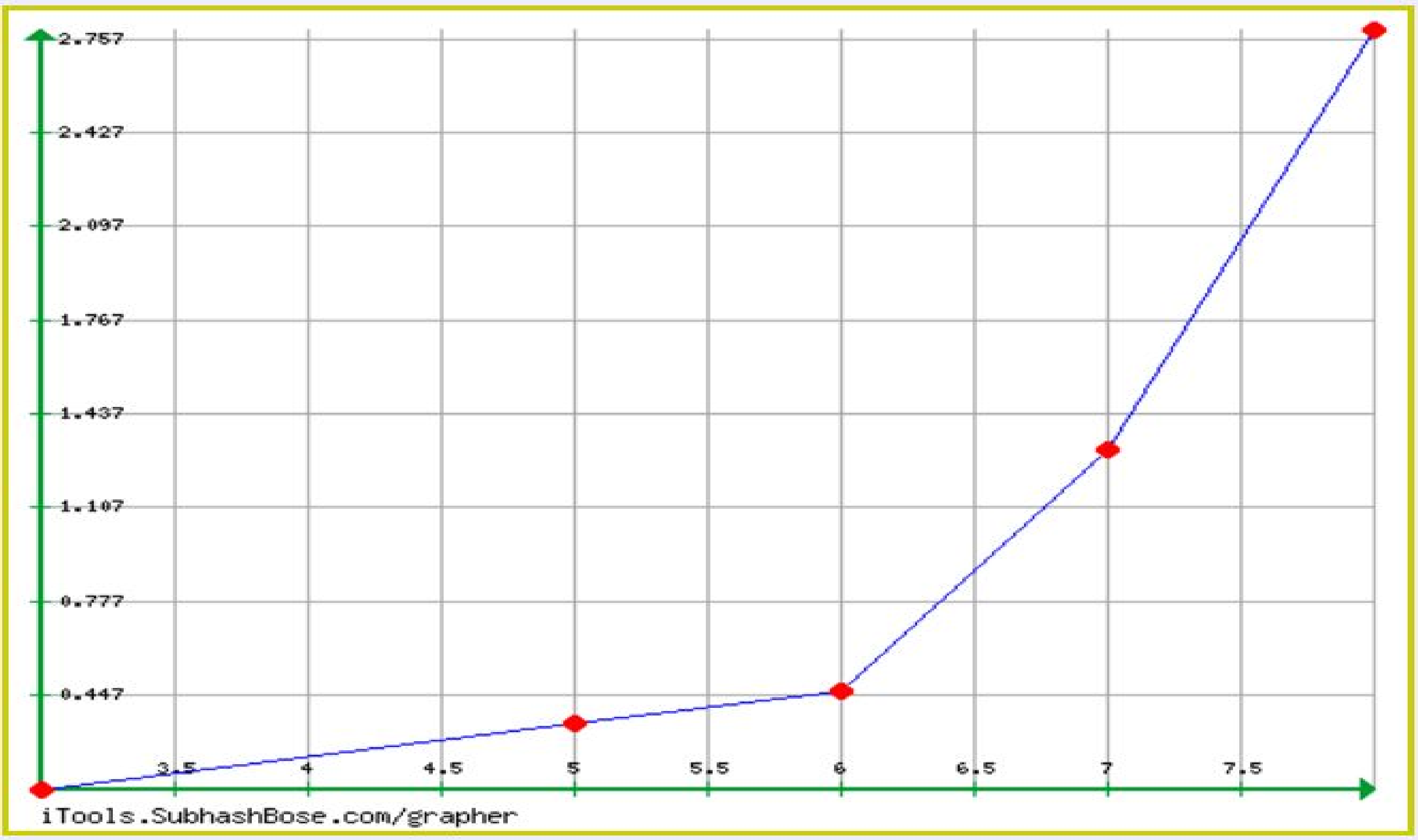
```
procedure CONSTRUCT(C)                ▷ where C = Cycle Space
    V ← empty set
    E ← empty set
    for each sub in C do
        for (each edge in sub) do
            x ← edge.source
            y ← edge.destination
            if x > y then
                swap(x, y)
            if edge between x and y not considered then
E = E ∪ edge(x, y)
            if x not considered then V = V ∪ x
            if y not considered then V = V ∪ y
    return (V, E)
```

For Spanning Trees

Algorithm 2 Spanning Trees

```
procedure SPANNINGTREES(SP,EE,IND,E)
    if sp not already considered then
        Print(sp)
    if ind is greater than equal to size of E then
        Return
    SpanningTrees(sp, ee, ind + 1, E)
    x ← E[ind].source
    y ← E[ind].destination
    if x-y does not form a cycle in sp then sp = sp ∪ (x − y)
        SpanningTrees(sp, ee − 1, ind + 1, E)
procedure MAIN(G(V,E))
    sp ← empty set
    np ← Number of vertices in V
    SpanningTrees(sp, np − 1, 0, E)
```

Number of Nodes vs Time Taken(ms) graph



Number of Nodes vs Time Taken(ms) graph

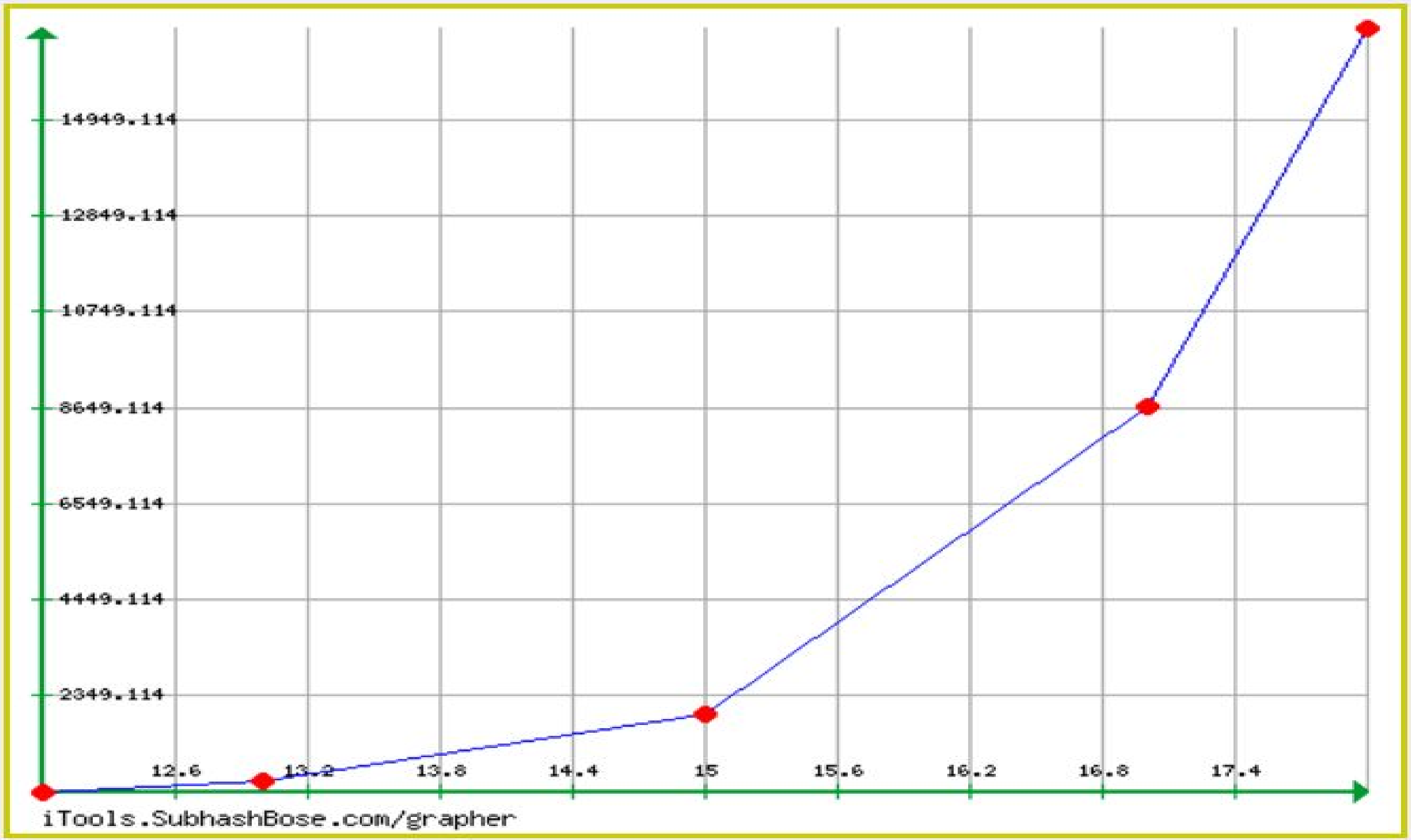


Table showing No. of nodes vs Time Taken

Number of Nodes	Time Taken(in ms)
3	0.117
5	0.3470
6	0.4600
7	1.3120
8	2.7910
9	3.1680
12	249.1140
13	482.5180
15	1938.6630
17	8681.6350
18	16978.010

Conclusion

- With the cycle space as an input, Original Graph has been constructed.
- After constructing the Graph, we found the possible spanning trees recursively.
- The Time Complexity of the proposed algorithm is $O(Log(V)*2^E)$.