ASSIGNMENT

Department of Computer Science & Engineering

TOPIC:

DYNAMIC PROGRAMMING SOLUTION FOR STEINER TREE

SUBMITTED BY

1. AATHIL TA B160345CS

Department of Computer Science & Engineering

Abstract:

An Algorithm for solving the steiner problem on an finite undirected graph is presented. This Algorithm computes the set of edges of minimum length needed to connect a specified set of 't' nodes. If entire graph contain 'n' node Algorithm takes

$$n^3/2 + 3^t n^2$$

" $n^3/2$ " time is for finding All pair shortest path and it can be discarded if there is shortest path matrix.

Our Algorithm exploits optimal substructure property. It will start from set of terminal taking each element from it forming a tree recursion until size equals 2 and built up remaining subset from that subset.

By using DP Approach we can avoid recalculation of repeated subproblem.

Scanned by CamScanner

Algorithm:

~~Steiner~~

Steiner-Tree (G, T)

{ T be set of terminals.

// base condition

for each $t \in T$ do

for each $v \in V$ do

ST[t][v] = dist(t, v);

|T|-2 for (m = 2 to m ≤ |T|) do

$\binom{|T|}{m}$ { Let X be subset of size m.

|V| for each $v \in V$ do

{ ST[X][v] = ∞ ;

|V| for each $u \in V$ do.

$2^{m}-1$ { for each non disjoint non empty subset combination of X do (X' and X'', $X' \cap X'' = \phi$)

{

Sum = min(sum, ST[X'][u] + ST[X''][u])

}

ST[X][v] = min(ST[X][v], sum + dist[v][u])

}

if(|X| == |T|)

return.

}

}.

}.

Running time :-

$$\sum_{m=2}^{|T|-1} \binom{|T|}{m}(2^m - 1)|V|^2 = 3^{|T|}|V|^2$$

∝ Running time $= 3^{|T|}|V|^2$ where $|T|$ ∈ no of terminal.

optimal decomposition property:

Let S be any steiner tree Connecting Y, where $Y \subseteq N$ is a subset of nodes of Graph $G = (N, A)$, and Let q be any node of Y. If Contain atleast 3 members then there exist a $p \in N$ and subset D of Y ST

D is proper subset of $Y - \{q\}$ and D nonempty.

S contain 3 disjoint set S_1 S_2 S_3.

S_1 Connect $\{p, q\}$ S_2 Connect $\{p\} \cup D$.

S_3 Connect $\{p\} \cup (Y - D - \{q\})$.

furthermore S_1 S_2 S_3 are all stiener path Connecting Respective set.

Recursive Algorithm:

```
ST ( G, T ).
{
    if (|T| = = 2)
      { let t' and t'' be two elements.
          for each v ∈ G.V do
              return min ( d(v, t') + d(v, t'')
      }
    else
      { for each t ∈ T  do
            for each v ∈ G.V  do
            {
                return min ( ST(G, T-t0) + d(v, t0))
            }
      }.
}.
```

Running time of Recursive Solution = $|T|^{|T|} \times n$.

PROOF:

The proof goes by dynamic programming.

Pick any terminal 't_0' and let $T' = T \setminus \{t_0\}$.

For every nonempty $X \subset T'$ and every $v \in V$ we compute:

$ST(X,v)$ = minimum edge weight of a Steiner tree for $(X \cup \{v\})$

Note that we allow $v \in X$ The answer is stored in $ST(T', t_0)$

1-The trivial case: If $X=\{x\}$ for some $x \in T'$

then for every $v \in V$ we set

$ST(\{x\},v) = dist_G(x,v)$.

2-Now suppose $|X| \geq 2$ Look at the tree from v Starting from v go along the tree until you reach either a vertex in X or a vertex of degree at least

3. Let us call it u. Possibly u=v.

If $u \in X$ then we let $X'=\{u\}$.

Otherwise we let X' be the vertices in X in one connected component of the tree with {u} removed.In both cases we have $\emptyset != X'$ (X and the tree can be split into three pieces

- the path fromv to u(possibly trivial)
- a tree for u and X'(possibly trivial)
- a tree for u and X\X

$ST(X,v) = \min_{v \in V}(dist_G(v,u) + (\text{for all subset } x')\min(ST(X',u) + ST(X \setminus X',u)))$

Running time:

Each vertex of T'can be either in X', in $X \setminus X'$, or in $T' \setminus X$.

There are $3^{t-1} * n^2$ evaluations of the recurrence

Correctness of Algorithm:

Let T be set containing Terminals, V be set of All vertices, E be edges of the given graph.

```
steiner-Tree ( G, T)
{
  for each  t ∈ T  do
     for each v ∈ V
         ST[t][v] = dist(t, v)

  for (m = 2  to  m ≤ |T| )  do            —— (A)

   {    for each  subset X  of size 'm' do    — (B)
             for each u ∈ V
             {  ST[X][V] = ∞

                 for each  U ∈ V  do
 (C)  <———————  {   for each non disjoint non empty
                      subset Combination of X do
                    {  let X', X'' disjoint subset.
                       Sum = min ( sum, ST[X'][U] +
                                        ST[X''][U] ).
                    }
                 ST[X][V] = min ( ST[X][V] , sum+dist
             }.
        If |X| == |T|
            return  ST[X][last terminal].
   }.
```

Loop Invariant: at the start of each iteration of A we have ~~..~~ minimum stiener wieght for all terminal ~~...~~ subset of size |m-1|.

Initialization: Just before first iteration of loop A we have all one size terminal set stienerer wieght ~~..~~ value in ST[][] array [from base case].

Maintanance: Inorder to prove loop invariant, we need to take a look inside each loops inside loop A. In loop B we are generating all 'm' size subset, so All ~~a~~ 'm' ~~...~~ size subset are generated. inside that loop B we are generating All disjoint non empty subset of X [x' and x'', $x' \cap x'' = \phi$, $x' \cup x'' = x$}, $x', x'' \neq \phi$]. since by optimal substructure property the $ST[x'][v]$ and $ST[x''][v]$ will be already filled in table $\because |x'| \& |x''| < m$. so in loop C we will take minimmm value from all subset and fill the $ST[X][V]$ from most optimum value with help of All pair shortest path graph

~~...~~ " $\therefore$ after each ~~..~~ the i^{th} iteration of loop A we have filled up ~~...~~ $(i+2-1)^{th}$ size subset. value ~~to~~ in ST[][] ~~...~~ matrix.

Termination: when $m = |T|+1$ loop A All ~~...~~ value in ~~subset~~ ST[][] graph has been filled and ST[~~..~~ T][T.least()] will return minimmm value of stiener tree.

In short,

loop (A) will emulate all 'm' sized subset of T.

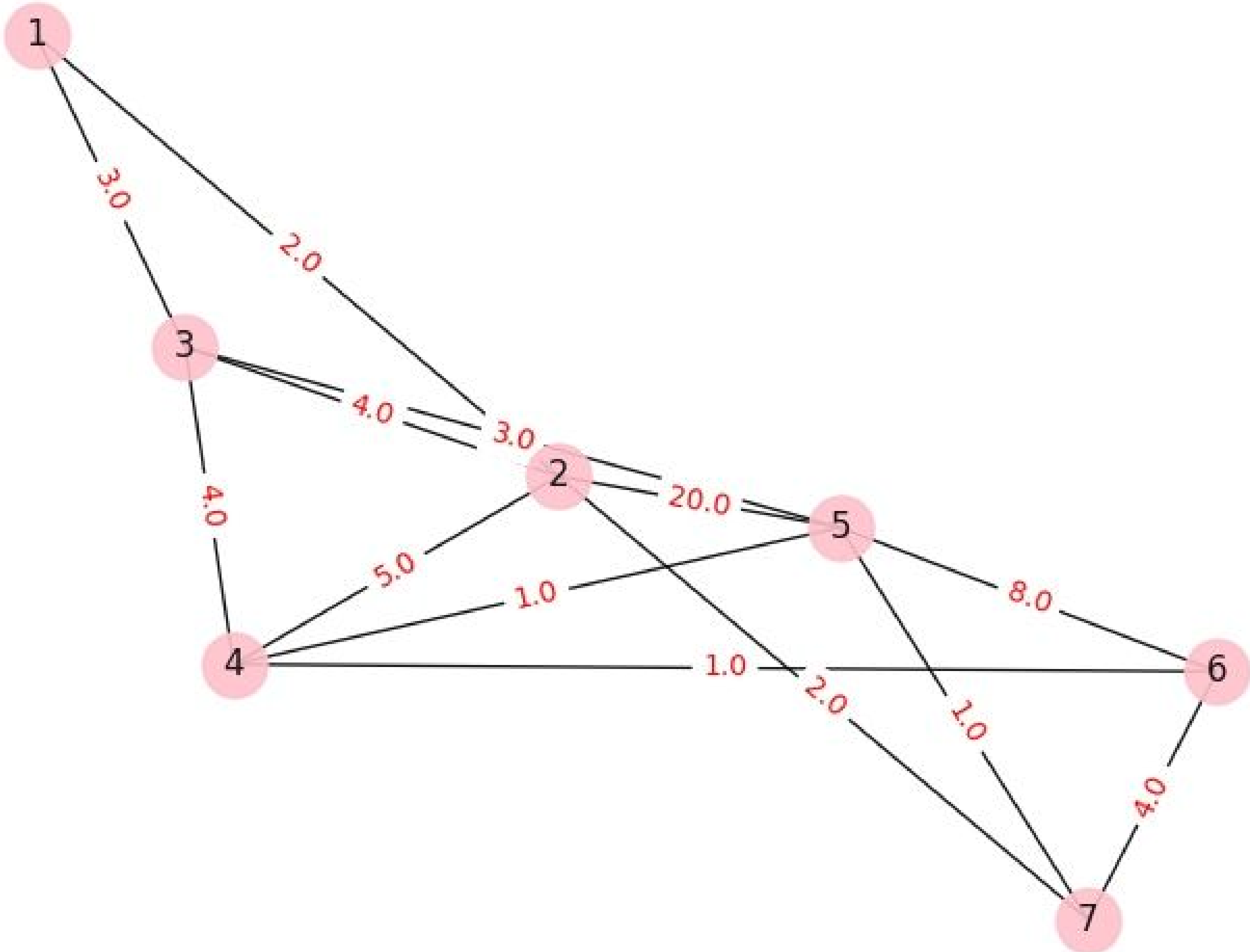
But due to optimal substructure property we have Steiner table filled for all subset of T of size less than m.

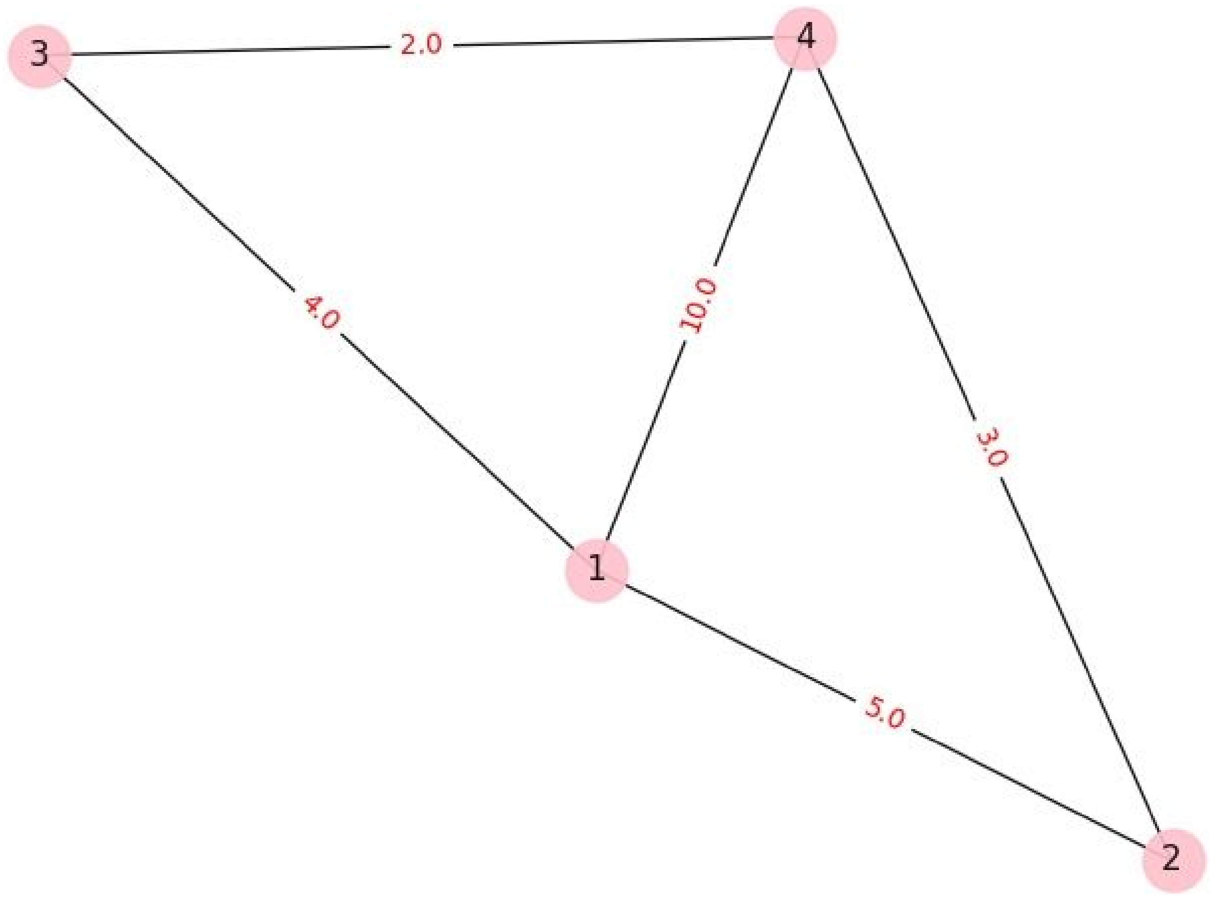
loop (B) will generate all subset of taken size m. thus loop (B) ensure filling of $^{|T|}C_m$ subset in ST[][] table.

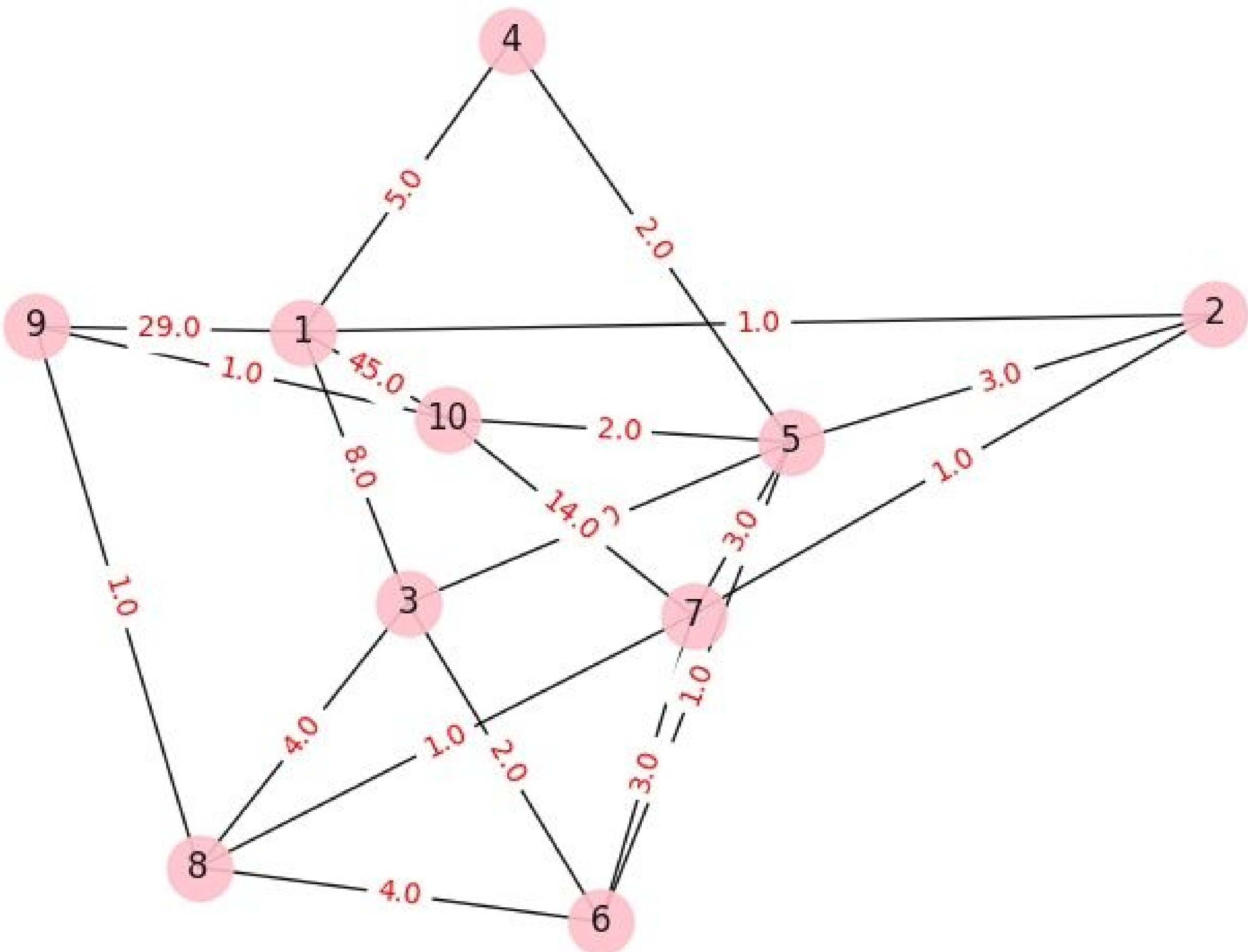
loop (C) will enumerate all subset of subset taken from loop (B) and filled ~~table~~ ST[][] table from all possible ~~subset~~ disjoint non empty subset part~~given~~tion of given input X. minimum value is taken from all posible enumeration.

- filling of DP table is bottom up By using fact of optimal decomposition property of given problem.

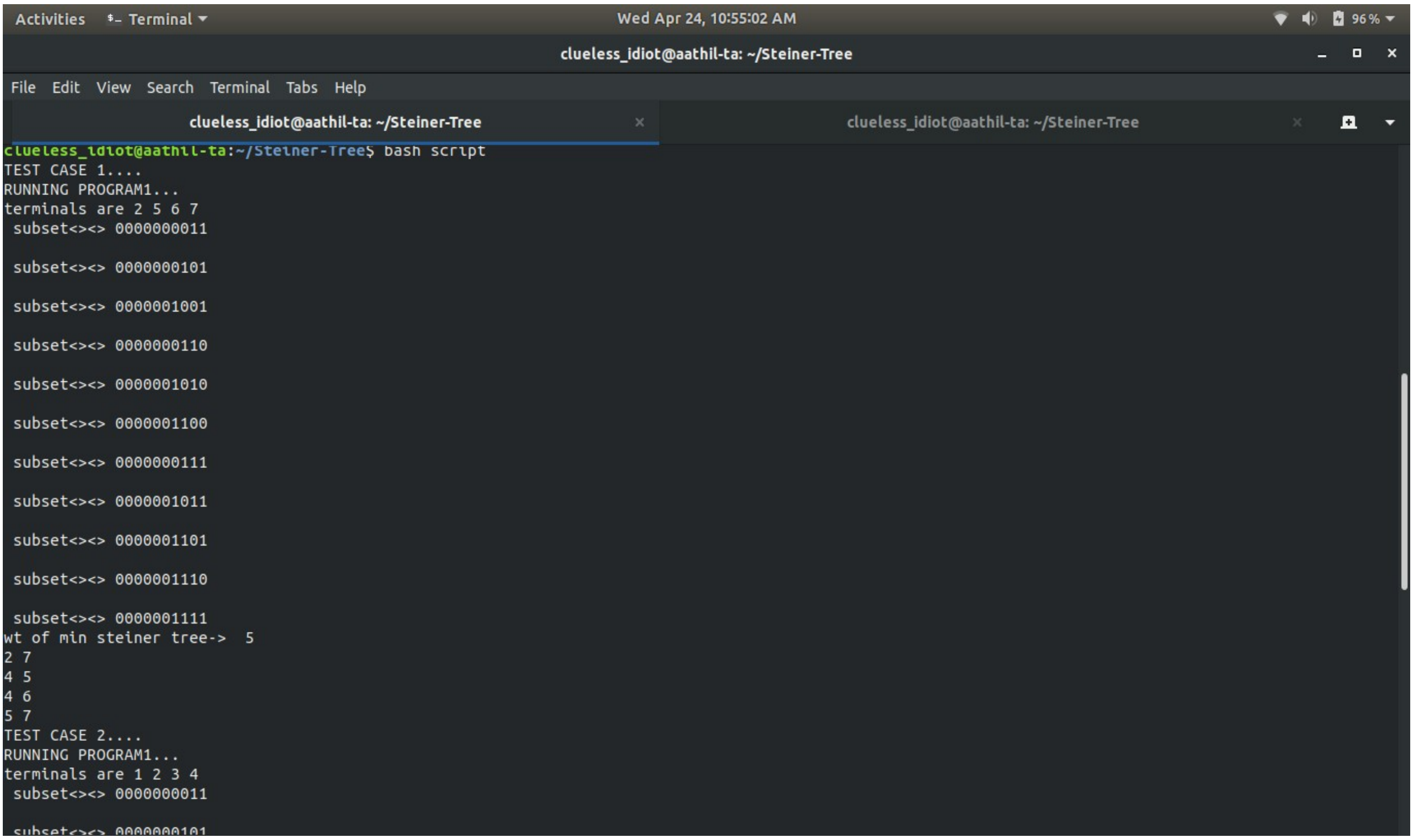
TEST CASES

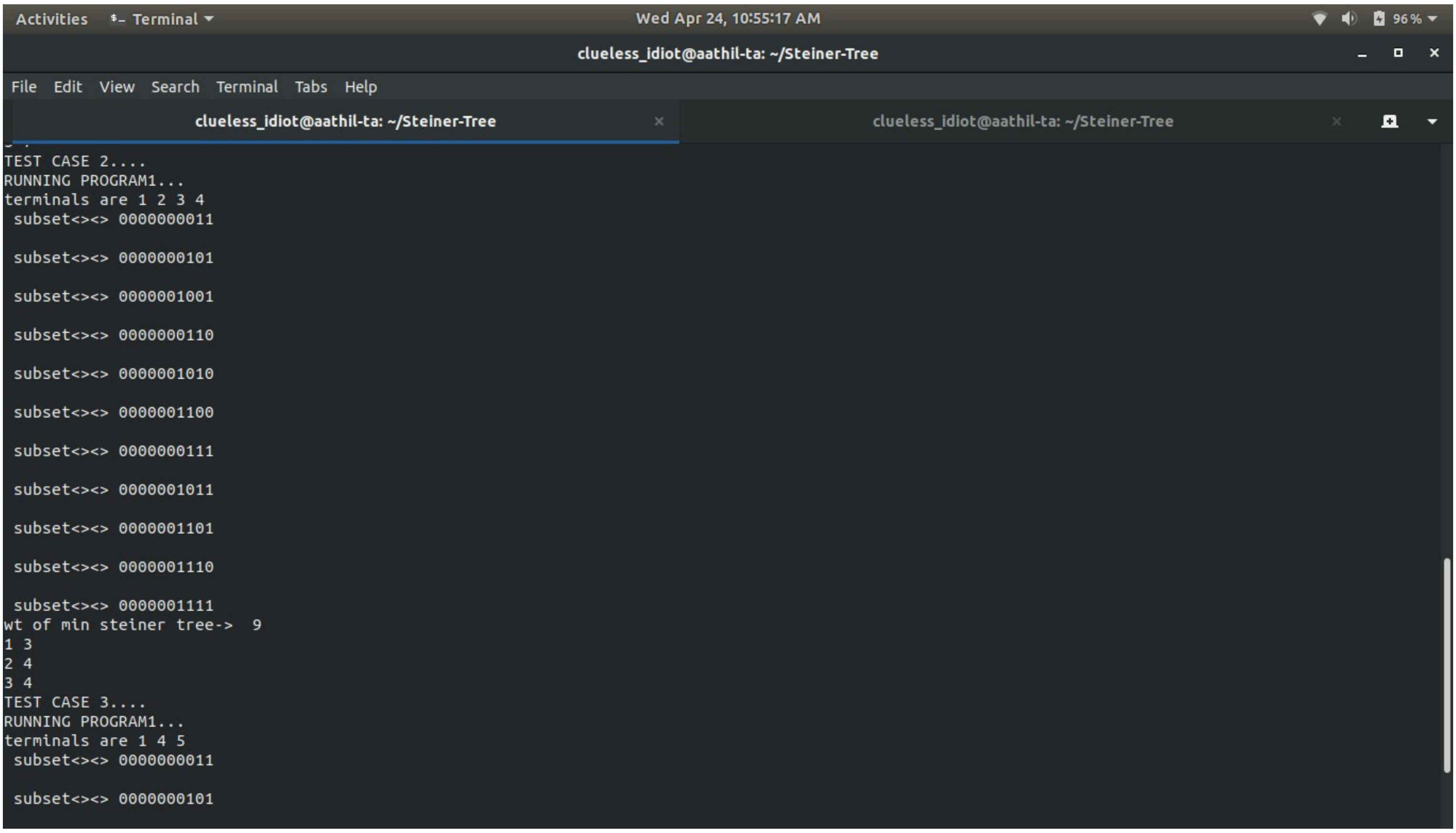


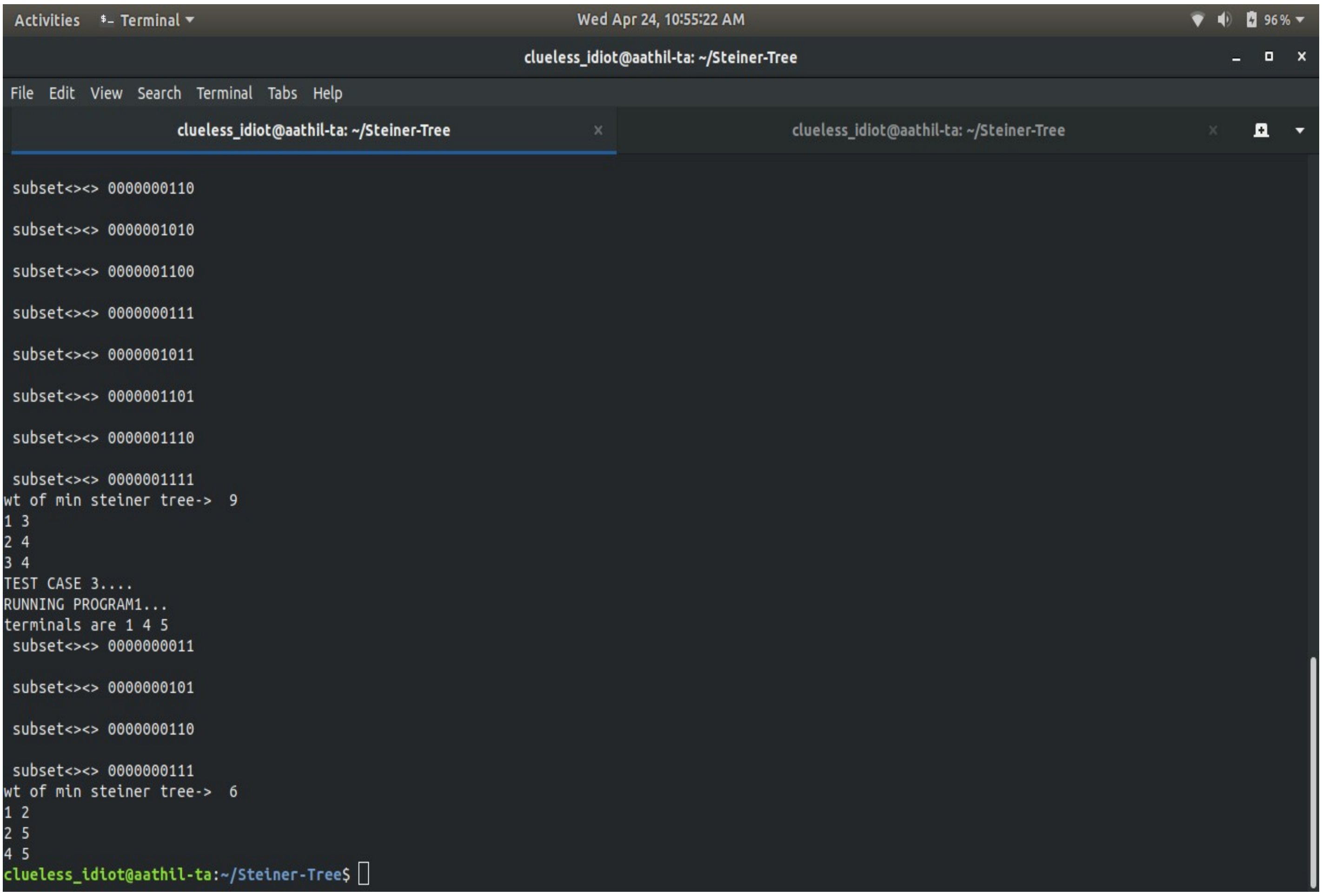

3
2.0
4
4.0
10.0
3.0
1
5.0
2


4
9
1
2
10
5
3
7
8
6
5.0
2.0
29.0
1.0
45.0
1.0
3.0
2.0
8.0
1.0
14.0
3.0
1.0
1.0
4.0
1.0
2.0
3.0
1.0
4.0

OUTPUT SCREENSHOTS






Activities
Terminal
Wed Apr 24, 10:55:22 AM
96 %
clueless_idiot@aathil-ta: ~/Steiner-Tree
File Edit View Search Terminal Tabs Help
clueless_idiot@aathil-ta: ~/Steiner-Tree
clueless_idiot@aathil-ta: ~/Steiner-Tree
subset<><> 0000000110
subset<><> 0000001010
subset<><> 0000001100
subset<><> 0000000111
subset<><> 0000001011
subset<><> 0000001101
subset<><> 0000001110
subset<><> 0000001111
wt of min steiner tree-> 9
1 3
2 4
3 4
TEST CASE 3....
RUNNING PROGRAM1...
terminals are 1 4 5
subset<><> 0000000011
subset<><> 0000000101
subset<><> 0000000110
subset<><> 0000000111
wt of min steiner tree-> 6
1 2
2 5
4 5
clueless_idiot@aathil-ta:~/Steiner-Tree$