# ASSIGNMENT

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

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 ~~the~~ edges of minimum length needed to connect a specified set of 't' nodes. If entire graph contain 'n' node Algorithm takes

$$n^3/_2 + 3^t n^2$$

4 $n^3/_2$ time ~~for finding Algorithm~~ ~~shortest~~ path and it can be ~~disgarded~~ 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 ~~down~~ until size equals 2 and built up remaining subset from that subset.

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

## 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 $m \leq |T|)$ do

$\binom{|T|}{m}$    { let x be subset of size $m$.

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

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

$2^m-1$      { for each $n·m$ disjoint no 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 :-

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

$$\therefore \quad \text{Running time} = 3^{|T|} |v|^2 \quad \text{where } |T| \in \text{ no of term} \\ \text{inal.}$$

## Optimal decomposition property:

Let S be 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 D of Y ST

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

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 stienor 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 t₀ ∈ 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 't0' 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 $\varnothing \neq X'$ (X and the tree can be split into three pieces

- the path from v 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\X′, or in T′\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                    — Ⓐ
    {
        for each  subset X  of  size 'm' do         — Ⓑ
            for each v ∈ V
            {  ST[X][v] = ∞

                for each u ∈ V  do
                {
          Ⓒ ←————————  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] , sumt dist
            }
        }
    }

    if |X| == |T|
        return ST[X][last terminal].
}
```

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

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

**Maintenance:** In order to prove loop invariant, we need to take a look inside each loops inside loop Ⓐ. In loop Ⓑ we are generating all 'm' size subset, so All 'm' size subset are generated. In the loop Ⓒ we are generating All disjoint non empty subset of X [x' and x'', $x' \cap x'' = \phi$, $x' \cup x'' = x$], x', x ≠ φ]. since by optimal sub structure property the ST[x'][v] and ST[x''][v] will be already filled in table ∵ |x'| & |x''| < m. so in loop Ⓒ we will take minimum value from all subset and fill the ST[x][v] from most optimum value with help of All pair shortest path graph.

**Termination** ∴ after each iᵗʰ the iteration of loop Ⓐ we have filled up (i+2-1)ᵗʰ size subset. value in ST[][] matrix.

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

In short,

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

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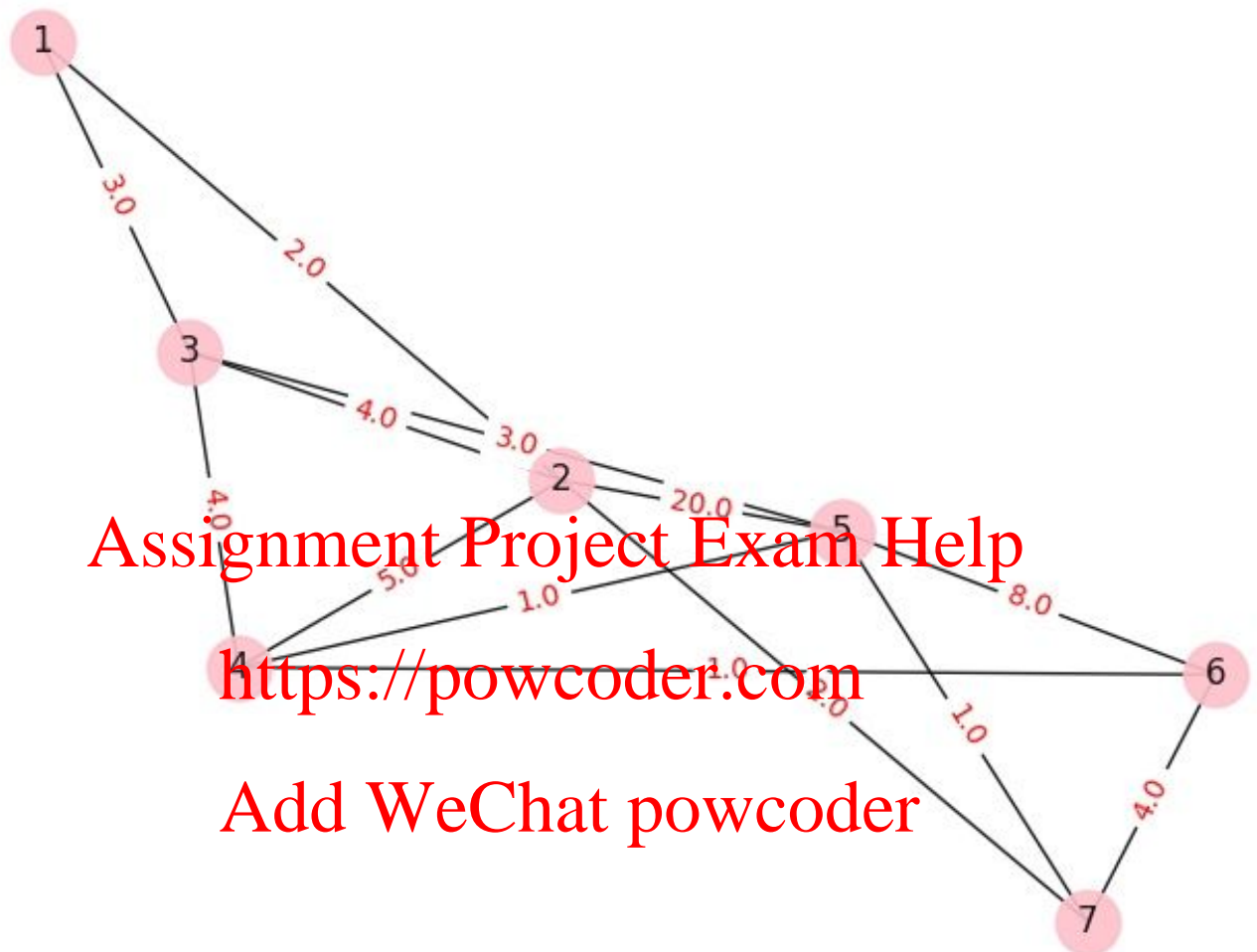than m.

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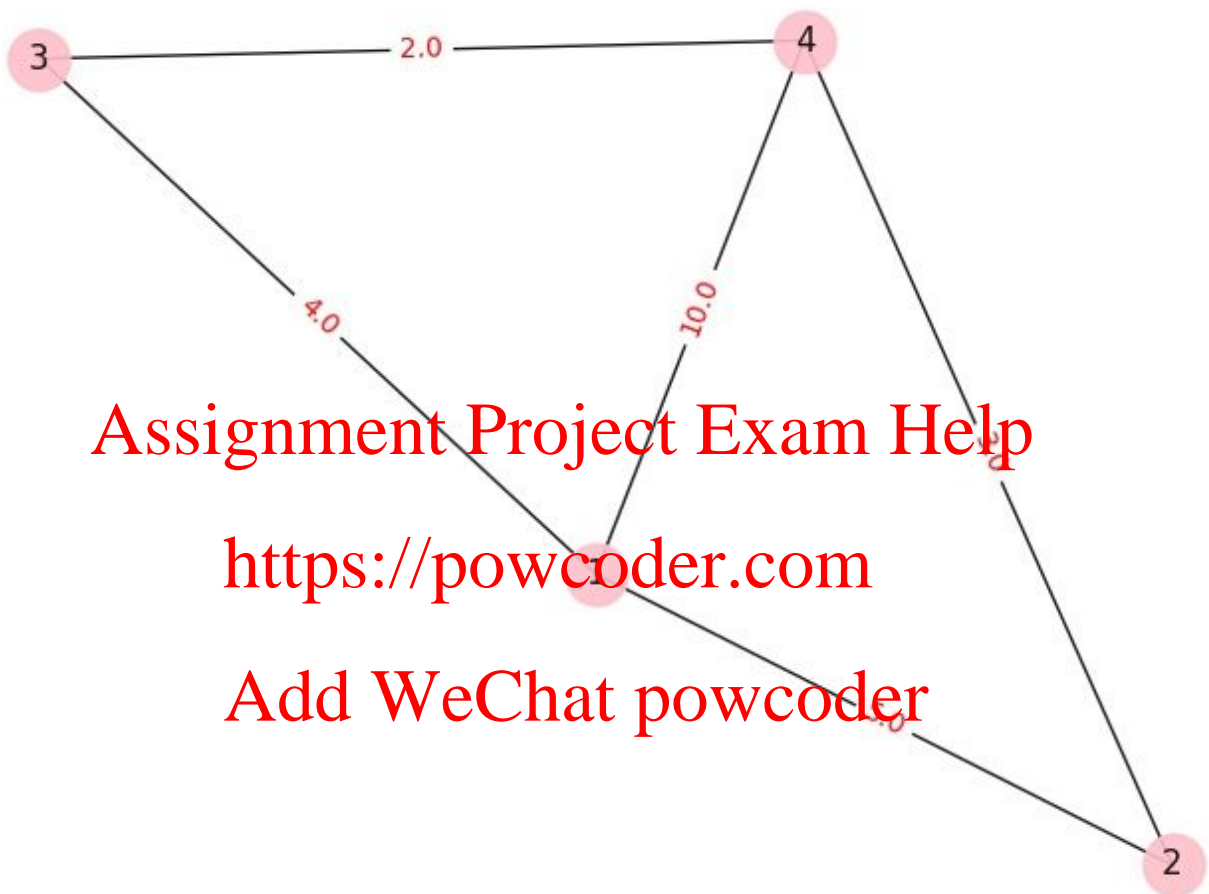thus loop Ⓑ ensure filling of all $^{|T|}C_m$ subset in
ST[][] table.

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

• filling of DP table is bottom up By using fact of optional
decomposition property of given problem.
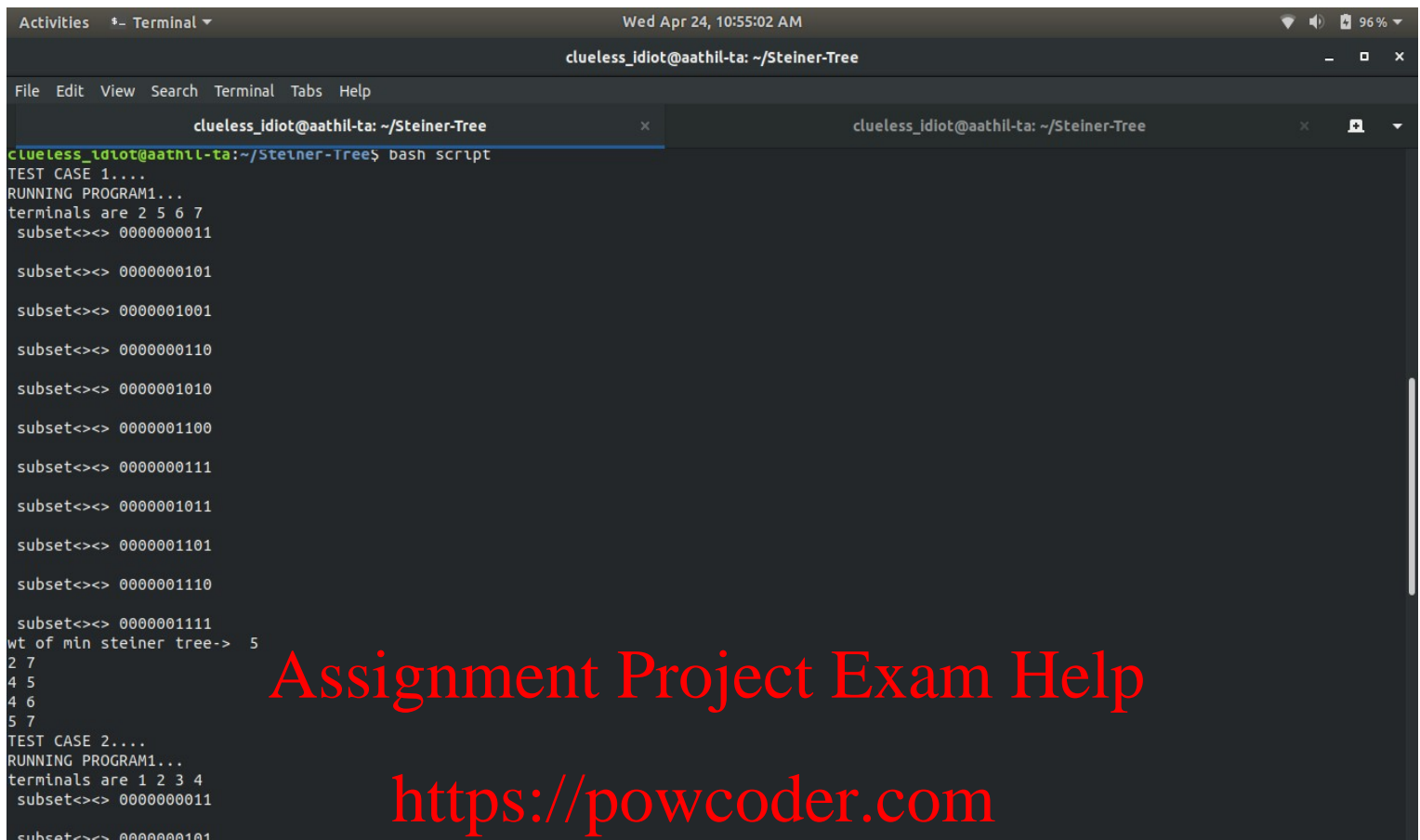
3 ———— 2.0 ———— 4

4.0

10.0

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

2

OUTPUT SCREENSHOTS

Activities    Terminal                                    Wed Apr 24, 10:55:02 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          ×
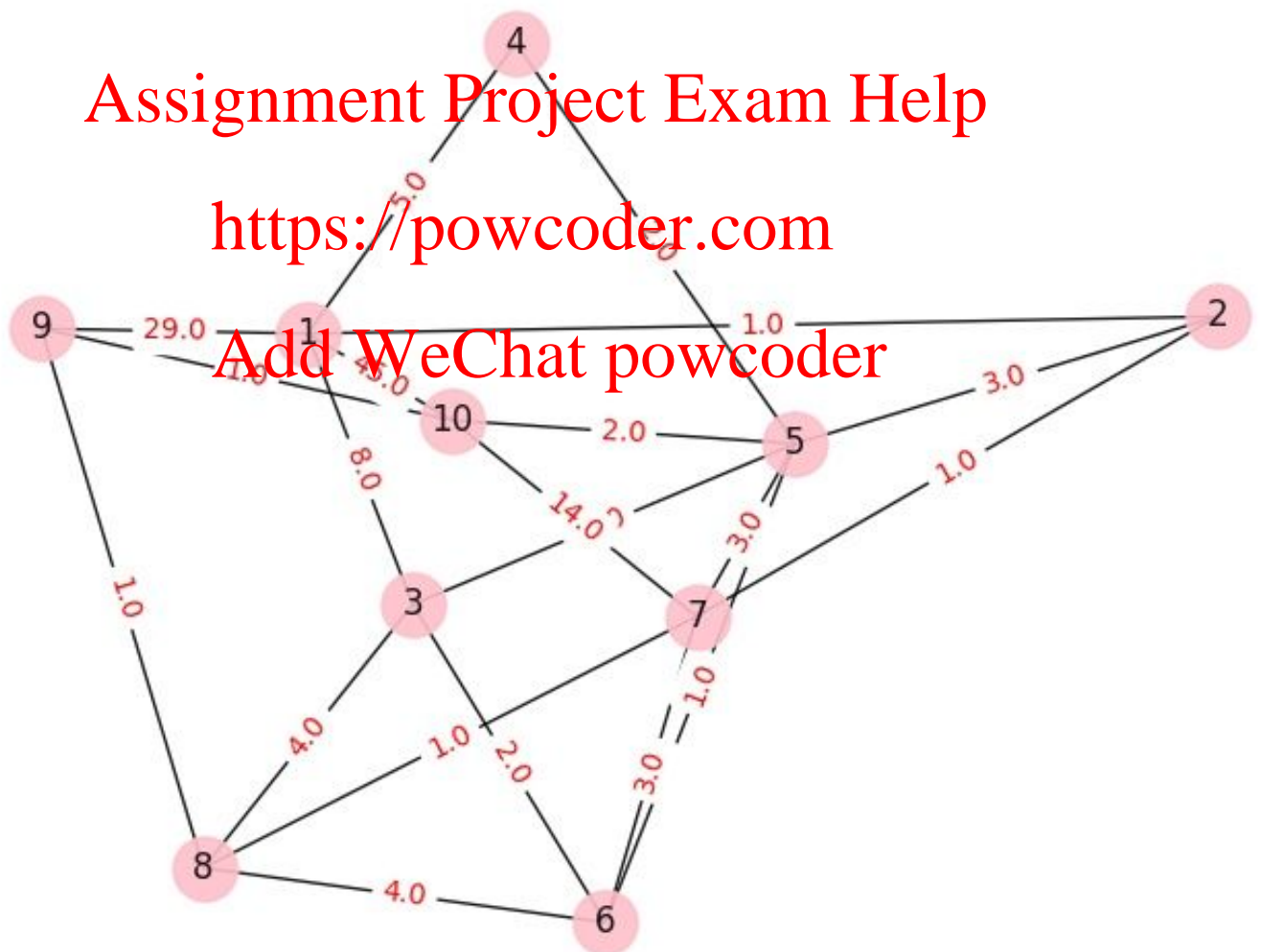
```
clueless_idiot@aathil-ta:~/Steiner-Tree$ bash script
TEST CASE 1....
RUNNING PROGRAM1...
terminals are 2 5 6 7
 subset<><> 0000000011

 subset<><> 0000000101

 subset<><> 0000001001

 subset<><> 0000000110

 subset<><> 0000001010

 subset<><> 0000001100

 subset<><> 0000000111

 subset<><> 0000001011

 subset<><> 0000001101

 subset<><> 0000001110

 subset<><> 0000001111
wt of min steiner tree-> 5
2 7
4 5
4 6
5 7
TEST CASE 2....
RUNNING PROGRAM1...
terminals are 1 2 3 4
 subset<><> 0000000011
 subset<><> 0000000101
```
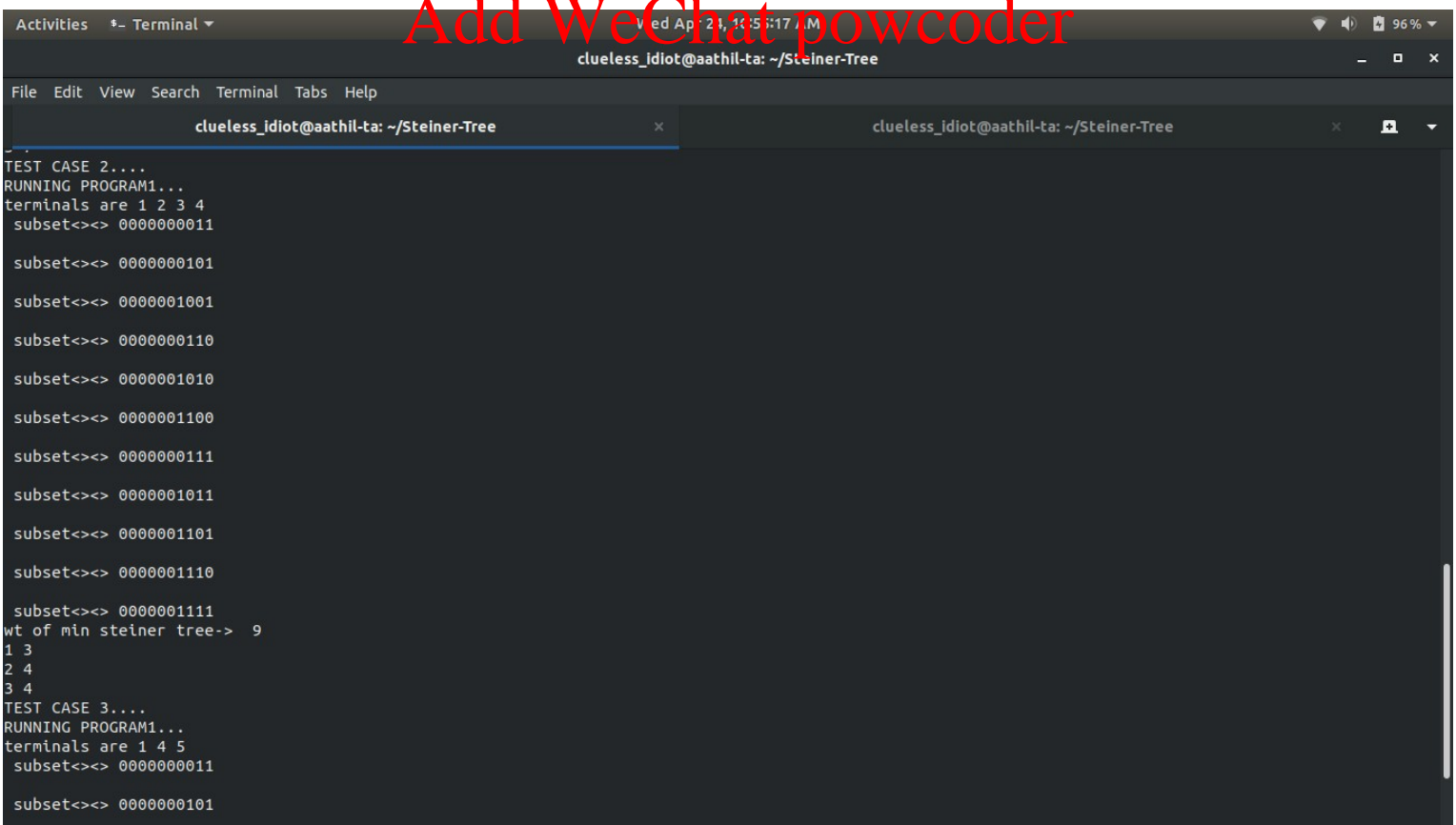
Activities    Terminal                                    Wed Apr 24, 10:55:17 AM                                      96%

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          ×

```
TEST CASE 2....
RUNNING PROGRAM1...
terminals are 1 2 3 4
 subset<><> 0000000011

 subset<><> 0000000101

 subset<><> 0000001001

 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
```

clueless_idiot@aathil-ta: ~/Steiner-Tree                                                      _  □  x

File  Edit  View  Search  Terminal  Tabs  Help

| clueless_idiot@aathil-ta: ~/Steiner-Tree | x | clueless_idiot@aathil-ta: ~/Steiner-Tree | x | 🗗 ▼ |

```
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$ ▯
```