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 v ∈ V do

{ for each non disjoint non empty subset Combination of X do — C

{ let X', X'' disjoint subset.

~~Sum~~ Sum = min (sum, ST[X'][v] + ST[X''][U]).

}

ST[X][V] = min (ST[X][V], sum + dist

}.

if |X| = = |T|

return ST[X][~~V~~ last terminal].

}.

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~~ 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].
   }.
```

Scanned by CamScanner

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

Initialization: Just before first iteration of loop (A) we have all one size terminal set steinerer wieght ~~[illegible]~~ 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 ~~[illegible]~~ 'm' size subset are generated. inside that loop (B) we are generating All disjoint nonempty subset of X $[x' \text{ and } x'', x' \cap x'' = \phi$ $x' \cup x'' = x\}, x', x'' \neq \phi]$. 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 minimum value from all subset and fill the ST[X][V] from most optimum value with help of All pair shortest path graph

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

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

Loop Invariant: at the start of each iteration of A we have ~~ar~~ 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 ~~....~~ 'm' size subset are generated. inside that loop B we are generating All disjoint nonempty subset of X [x' and x'', $x' \cap x'' = \phi$ $x' \cup x'' = x$}, $x', x'' \neq \phi$]. since by optimal substructre 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 minimum 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 ~~.~~ ith the 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 ~~...~~ ~~subset~~ in ST[][] graph has been filled and ST[T][T.least()] will return minimum 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 ST[][] table from all possible distinct non empty subset partition 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.