Minimum Bottleneck Spanning Trees (MBST)

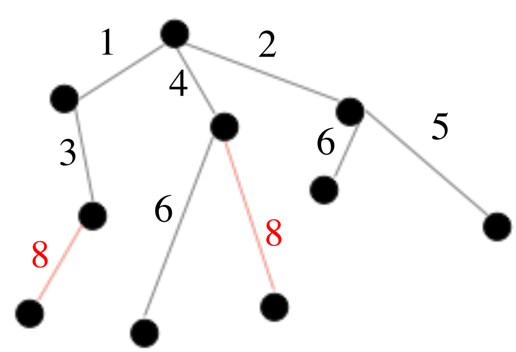
DONGFENG GU

Outline

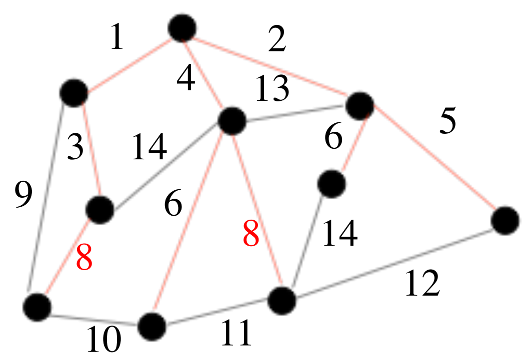
1. Definition
2. Relation between Minimum Bottleneck Spanning Tree (MBST) and Minimum Spanning Tree (MST)
3. Camerini’s algorithm for finding an MBST in an undirected connected graph
4. Camerini’s algorithm for finding an MBST in an directed connected graph

Bottleneck Edge

Bottleneck edges in a spanning tree are the edges with the maximum weight in the tree

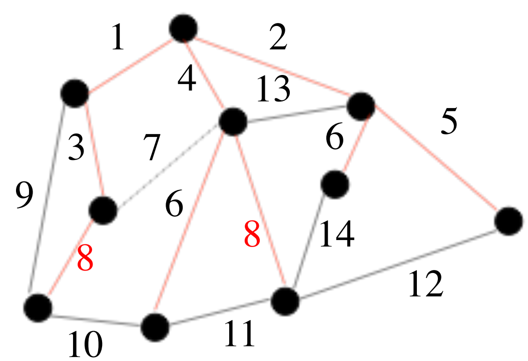


Minimum Bottleneck Spanning Tree (MBST)



MBST is a spanning tree which has the minimum bottleneck edge value among all the possible spanning trees in a graph G(V,E).

MBST VS MST

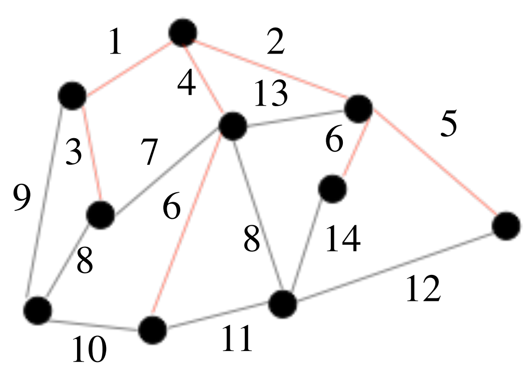


Minimum Spanning Tree (MST) is necessary a Minimum Bottleneck Spanning Tree (MBST) while the opposite is not.

Therefore any algorithm that can find the MST can also find the MBST.

Camerini's algorithm in undirected graph

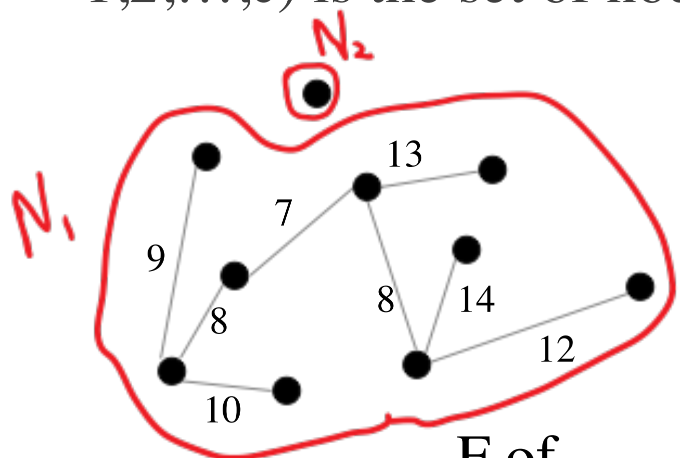
Given an undirected connected graph G(V,E). Let A be a subset of E such that W(e) >= W(e') for all e Є A', e'Є B = E\A. Let F be a maximal Forest of GB and η = N1 , N2 , N3 … Nc , where Ni (i = 1,2,…,c) is the set of nodes of the i-th component of F



Weight of A: 1,2,3,4,5,6,6
Weight of B: 7,8,8,9,10,11,12,13,14

Camerini's algorithm in undirected graph

Given an undirected connected graph G(V,E). Let A be a subset of E such that W(e) >= W(e') for all e Є A', e'Є B = E\A. Let F be a maximal Forest of GB and η = N1 , N2 , N3 … Nc , where Ni (i = 1,2,…,c) is the set of nodes of the i-th component of F



F of GB

Weight of A: 1,2,3,4,5,6,6
Weight of B: 7,8,8,9,10,11,12,13,14

Two theorems

Theorem 1: If F is a spanning tree of G, a Minimum Bottleneck Spanning Tree (MBST) of G is given by any MBST of GB .

Theorem 2: If F is not a spanning tree of G, a MBST of G can be obtained by adding to F any MBST of G’, where G’ is the graph GA collapsed in to η, i.e. G’ = (GA)η.

Pseudocode

Procedure 1 MBST(G,w)

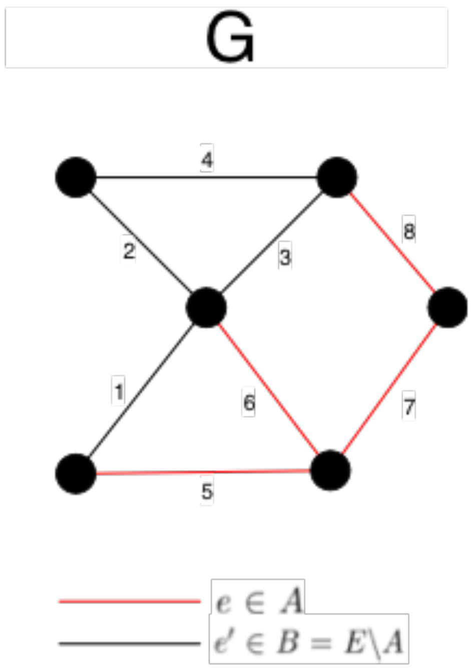
```
let E be the set of edges of G;
if |E| = 1 then
  return E
else
  A ← UH(E, w);
  B ← E\A;
  F ← FOREST(G_B);
  let η = N_1, N_2, ..., N_c, where N_i(i = 1, 2, ..., c) is the set of nodes of the i-th component of F;
  if c = 1 then
    return MBST(G_B, w)
  else
    return F ∪ MBST((G_A)_η, w);
  end if
end if
```

Pseudocode

Procedure 1 MBST(G,w)

```
let E be the set of edges of G;
if |E| = 1 then
    return E
else
    A ← UH(E, w);          |A| = ⌊|E|/2⌋
    B ← E\A;
    F ← FOREST(G_B);
    let η = N_1, N_2, ..., N_c, where N_i(i = 1, 2, ..., c) is the set of nodes of the i-th component of F;
    if c = 1 then
        return MBST(G_B, w)
    else
        return F ∪ MBST((G_A)_η, w);
    end if
end if
```

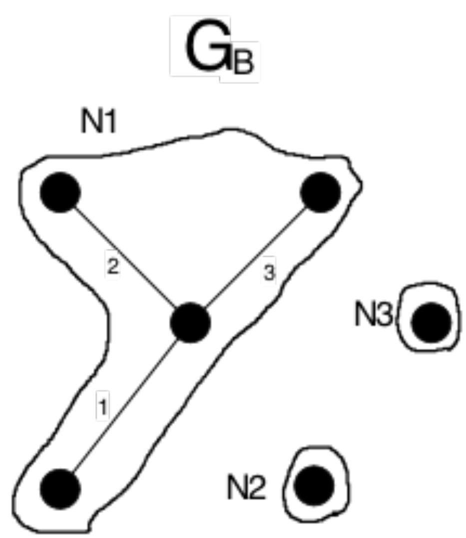
Example



Procedure 1 MBST(G,w)

```
let E be the set of edges of G;
if |E| = 1 then
  return E
else
  F ← FOREST(G_B);
  let η = N_1, N_2, ..., N_c, where N_i(i
  if c = 1 then
    return MBST(G_B, w)
  else
    return F ∪ MBST((G_A)_η, w);
  end if
end if
```

Example



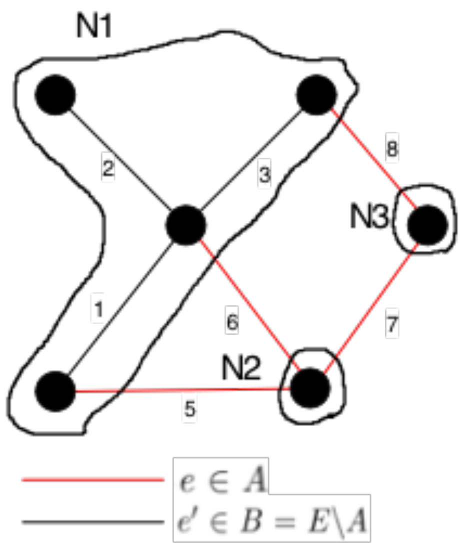
In this case, F is not a spanning tree of G

Procedure 1 MBST(G,w)

let E be the set of edges of G;
if $|E| = 1$ **then**
  return E
else
  $A \leftarrow UH(E, w)$;
  $B \leftarrow E \setminus A$;
  [illegible]
  let $\eta = N_1, N_2, ..., N_c$, where $N_i(i$
  if $c = 1$ **then**
    return $MBST(G_B, w)$
  else
    return $F \cup MBST((G_A)_\eta, w)$;
  end if
end if

Example

$(G_A)_\eta$

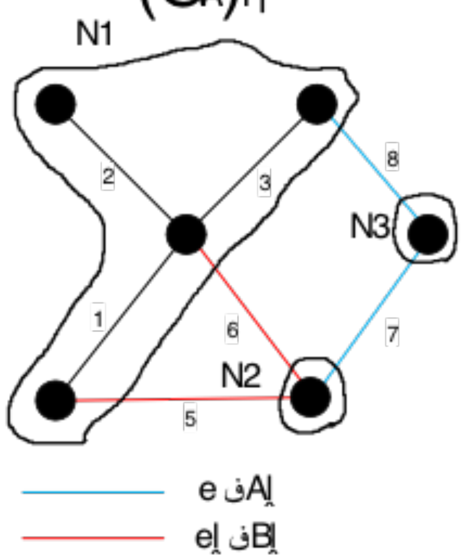


Procedure 1 MBST(G,w)

```
let E be the set of edges of G;
if |E| = 1 then
  return E
else
  A ← UH(E, w);
  B ← E\A;
  F ← FOREST(G_B);
  let η = N_1, N_2, ..., N_c, where N_i(i
  if c = 1 then
    return MBST(G_B, w)
  else
    return F ∪ MBST([illegible], w);
  end if
end if
```

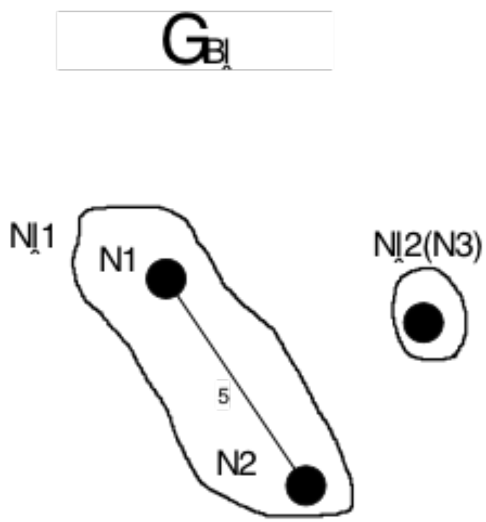
Example

MBST of $(G_A)_\eta$




```
Procedure 1 MBST(G,w)
  let E be the set of edges of G;
  if |E| = 1 then
    return E
  else
    F ← FOREST(G_B);
    let η = N_1, N_2, ..., N_c, where N_i(i
    if c = 1 then
      return MBST(G_B, w)
    else
      return F ∪ MBST((G_A)_η, w);
    end if
  end if
```

Example



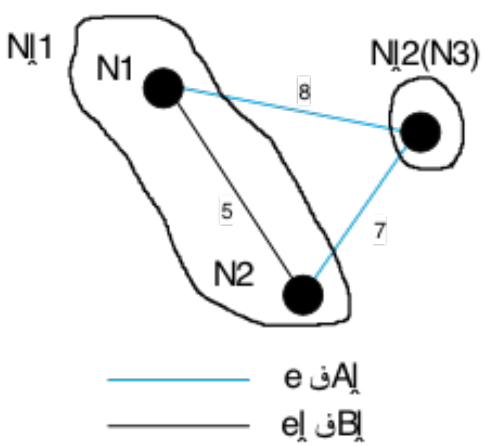
In this case, F is not a spanning tree of G

Procedure 1 MBST(G,w)

let E be the set of edges of G;
if $|E| = 1$ **then**
  return E
else
  $A \leftarrow UH(E, w)$;
  $B \leftarrow E \setminus A$;
  let $\eta = N_1, N_2, ..., N_c$, where $N_i(i$
  if $c = 1$ **then**
    return $MBST(G_B, w)$
  else
    return $F \cup MBST((G_A)_\eta, w)$;
  end if
end if

Example

$(G_A)\ \eta$



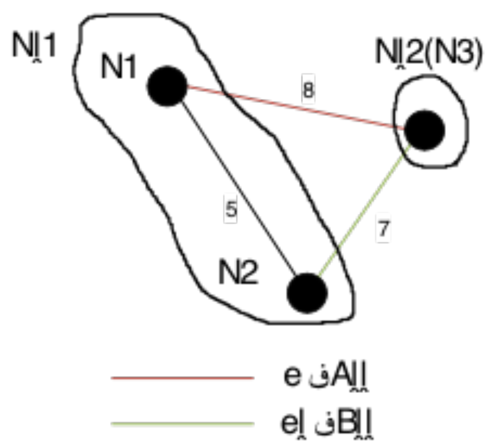
Procedure 1 MBST(G,w)

---

let E be the set of edges of G;
if $|E| = 1$ **then**
  return E
else
  $A \leftarrow UH(E, w)$;
  $B \leftarrow E \backslash A$;
  $F \leftarrow FOREST(G_B)$;
  let $\eta = N_1, N_2, ..., N_c$, where $N_i(i$
  if $c = 1$ **then**
    return $MBST(G_B, w)$
  else
    return $F \cup MBST$ $w)$;
  end if
end if

Example

MBST of $(G_A)_\eta$

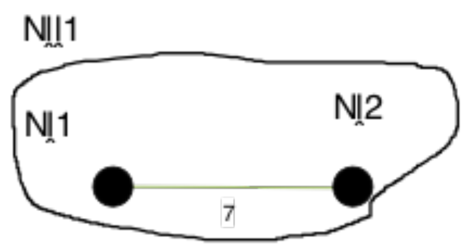


Procedure 1 MBST(G,w)

let E be the set of edges of G;
if $|E| = 1$ **then**
  return E
else
  $F \leftarrow FOREST(G_B)$;
  let $\eta = N_1, N_2, ..., N_c$, where $N_i(i$
  if $c = 1$ **then**
    return $MBST(G_B, w)$
  else
    return $F \cup MBST((G_A)_\eta, w)$;
  end if
end if

Example

G_B



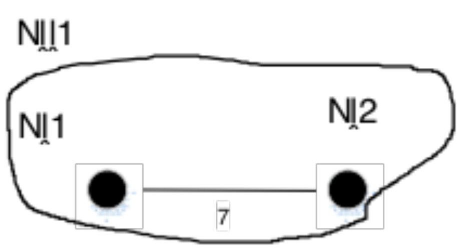
In this case, F is a spanning tree of G

Procedure 1 MBST(G,w)

```
let E be the set of edges of G;
if |E| = 1 then
  return E
else
  A ← UH(E, w);
  B ← E\A;
  let η = N_1, N_2, ..., N_c, where N_i(i
  if c = 1 then
    return MBST(G_B, w)
  else
    return F ∪ MBST((G_A)_η, w);
  end if
end if
```

Example

$MBST(G_{B||})$



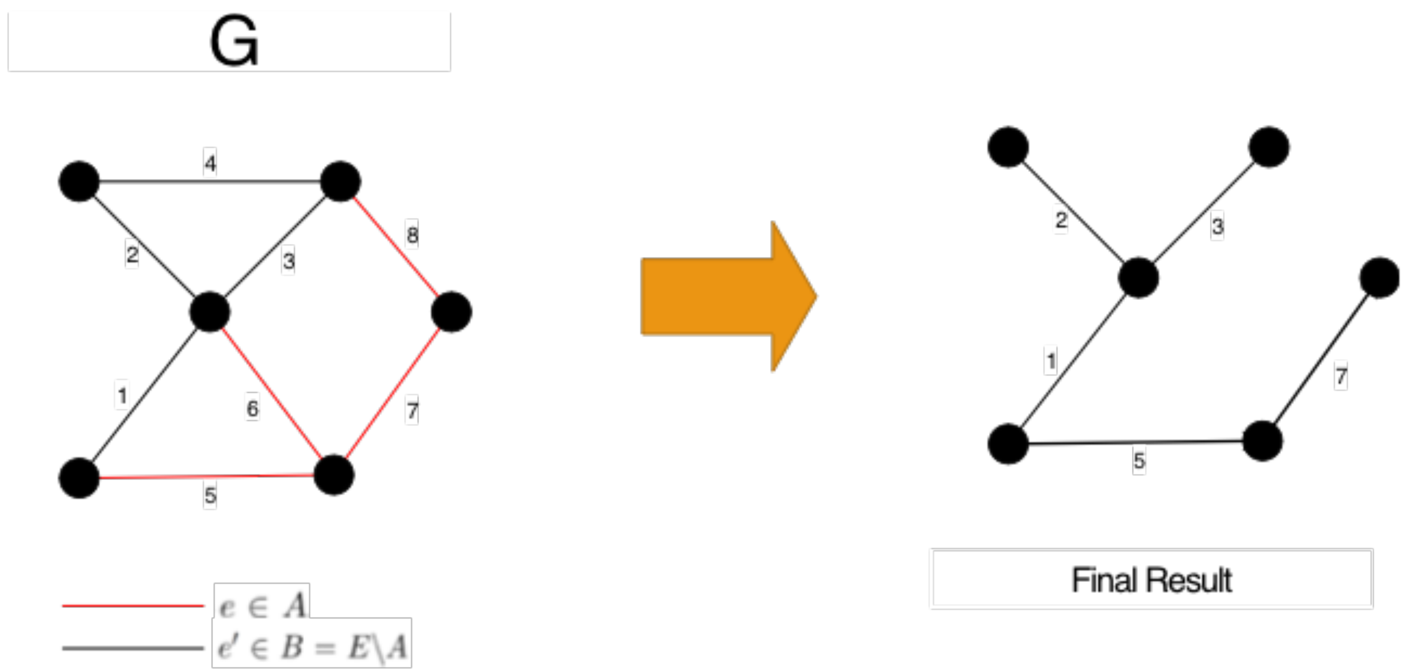
In this case, |E| = 1 return E

Procedure 1 MBST(G,w)

let E be the set of edges of G;
if $|E| = 1$ **then**

else
  $A \leftarrow UH(E, w)$;
  $B \leftarrow E \backslash A$;
  $F \leftarrow FOREST(G_B)$;
  let $\eta = N_1, N_2, ..., N_c$, where $N_i(i$
  if $c = 1$ **then**
    return
  else
    return $F \cup MBST((G_A)_\eta, w)$;
  end if
end if

Example



Time Complexity

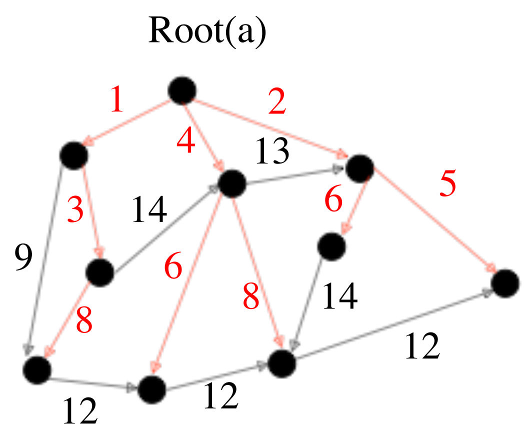
UH, FOREST all require
where m is the numbe

- Since UH is similar to f
them splitting the edge
median, and finding th

Procedure 1 MBST(G,w)

```
let E be the set of edges of G;
if |E| = 1 then
    return E
else
    A ← [illegible]
    B ← E \ A;
    F ← [illegible]
    let η = N_1, N_2, ..., N_c, where N_i(i
    if c = 1 then
        return MBST(G_B, w)
    else
        return F ∪ MBST((G_A)_η, w);
    end if
end if
```

Spanning arborescence (MBSA)



An arborescence of G(V,E) is a minimal subgraph of G which contains a directed path from a specified node “a” in G to each node of a subset V’ of V-{1}.

Node “a” is called the *root of* the arborescence. An arborescence is a spanning arborescence if V’ = V - {1}.

Camerini's algorithm for MBSA

1. T represents a
it is know that G_T
spanning arbores
node "a". Initially

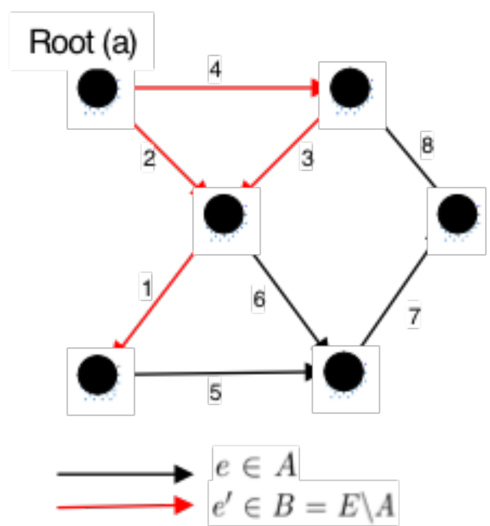
2. UH takes (E-T)
returns $A \subseteq$ (E-T)

Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
  A ← UH(E − T);
  B ← (E − T)\A;
  F ← BUSH(G_{B∪T});
  if F is a spanning arborescence of G then
    S ← F;
    MBSA((G_{B∪T}, w, T));
  else
    MBSA(G, w, T ∪ B);
  end if
end if
```

Example

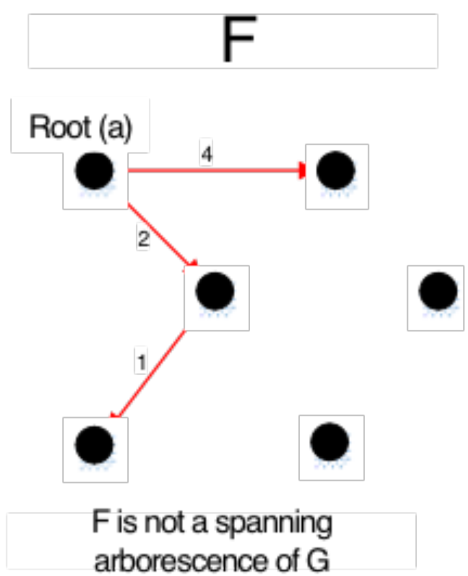
MBSA(G,w,T)



Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
  A ← UH(E − T);
  B ← (E − T)\A;
  F ← BUSH(G_{B∪T});
  if F is a spanning arborescence of G then
    S ← F;
    MBSA((G_{B∪T}, w, T));
  else
    MBSA(G, w, T ∪ B);
  end if
end if
```

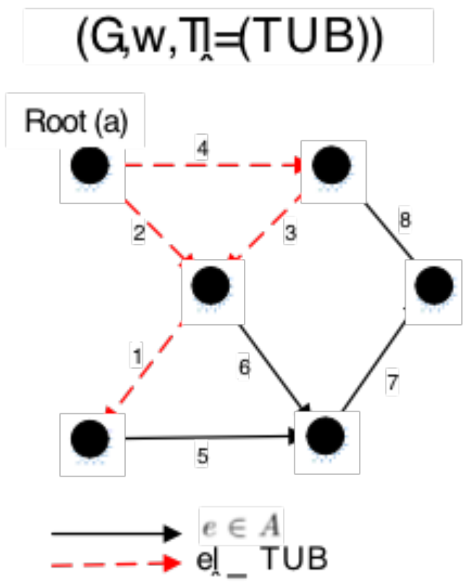
Example



Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
  A ← UH(E − T);
  B ← (E − T)\A;
  F ← BUSH(G_{B∪T});
  if F is a spanning arborescence of G then
    S ← F;
    MBSA((G_{B∪T}, w, T));
  else
    MBSA(G, w, T ∪ B);
  end if
end if
```

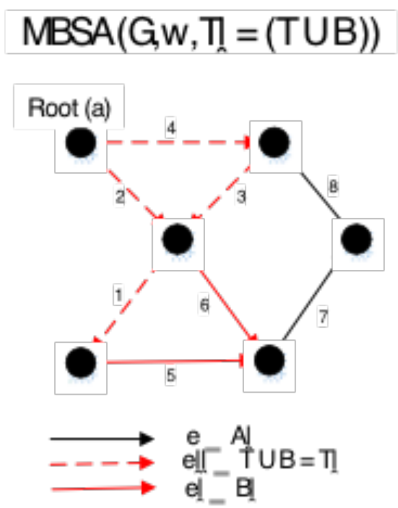
Example



Procedure 2 MBSA(G,w,T)

let E be the set of edges of G;
if $|E - T| > 1$ **then**
  $A \leftarrow UH(E - T)$;
  $B \leftarrow (E - T) \setminus A$;
  $F \leftarrow BUSH(G_{B \cup T})$;
  if F is a spanning arborescence of G **then**
    $S \leftarrow F$;
    $MBSA((G_{B \cup T}, w, T))$;
  else
    $MBSA(G, w, T \cup B)$;
  end if
end if

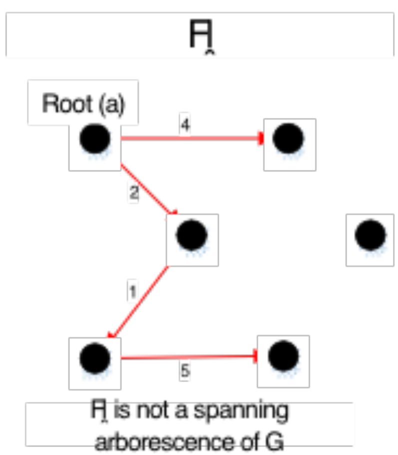
Example



Procedure 2 MBSA(G,w,T)

let E be the set of edges of G;
if $|E-T| > 1$ **then**
  $A \leftarrow UH(E-T)$;
  $B \leftarrow (E-T)\backslash A$;
  $F \leftarrow BUSH(G_{B\cup T})$;
  if F is a spanning arborescence of G **then**
    $S \leftarrow F$;
    $MBSA((G_{B\cup T}, w, T))$;
  else
    $MBSA(G, w, T \cup B)$;
  end if
end if

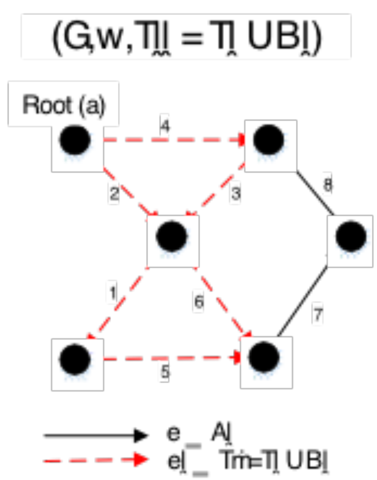
Example



Procedure 2 MBSA(G,w,T)

let E be the set of edges of G;
if $|E - T| > 1$ **then**
  $A \leftarrow UH(E - T)$;
  $B \leftarrow (E - T) \backslash A$;
  $F \leftarrow BUSH(G_{B \cup T})$;
  if F is a spanning arborescence of G **then**
    $S \leftarrow F$;
    $MBSA((G_{B \cup T}, w, T))$;
  else
    $MBSA(G, w, T \cup B)$;
  end if
end if

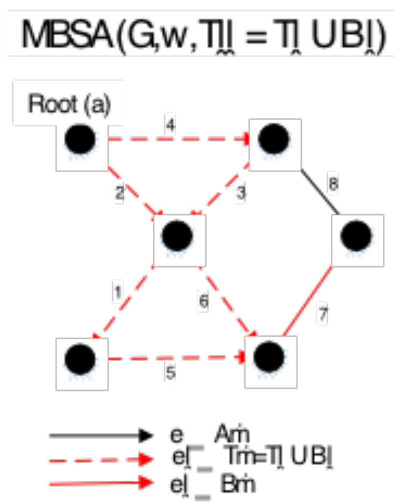
Example



Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
   A ← UH(E − T);
   B ← (E − T)\A;
   F ← BUSH(G_{B∪T});
   if F is a spanning arborescence of G then
      S ← F;
      MBSA((G_{B∪T}, w, T));
   else
      MBSA(G, w, T ∪ B);
   end if
end if
```

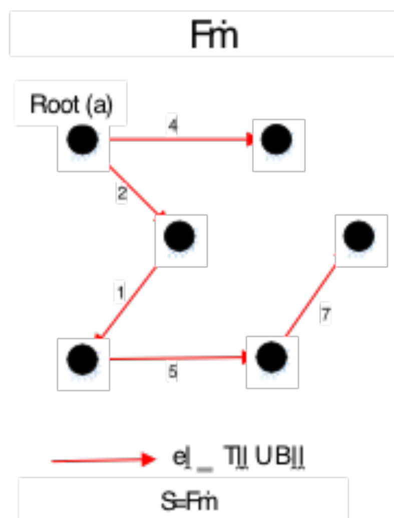
Example



Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
   A ← UH(E − T);
   B ← (E − T)\A;
   F ← BUSH(G_{B∪T});
   if F is a spanning arborescence of G  then
      S ← F;
      MBSA((G_{B∪T}, w, T));
   else
      MBSA(G, w, T ∪ B);
   end if
end if
```

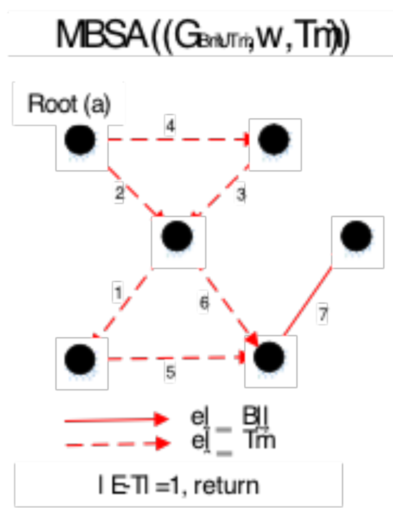
Example



Procedure 2 MBSA(G,w,T)

let E be the set of edges of G;
if $|E - T| > 1$ **then**
  $A \leftarrow UH(E - T)$;
  $B \leftarrow (E - T)\backslash A$;
  $F \leftarrow BUSH(G_{B \cup T})$;
  if F is a spanning arborescence of G **then**
    $S \leftarrow F$;
    $MBSA((G_{B \cup T}, w, T))$;
  else
    $MBSA(G, w, T \cup B)$;
  end if
end if

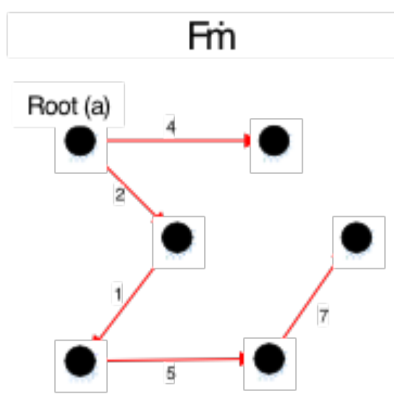
Example



Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
  A ← UH(E − T);
  B ← (E − T)\A;
  F ← BUSH(G_{B∪T});
  if F is a spanning arborescence of G then
    S ← F;
    MBSA((G_{B∪T}, w, T));
  else
    MBSA(G, w, T ∪ B);
  end if
end if
```

Example



Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
  A ← UH(E − T);
  B ← (E − T)\A;
  F ← BUSH(G_{B∪T});
  if F is a spanning arborescence of G then
    S ← F;
    MBSA((G_{B∪T}, w, T));
  else
    MBSA(G, w, T ∪ B);
  end if
end if
```

Time Complexity

UH require O(m)

BUSH requires O(m) at each execution and the number of these executions is O(logm) since |E-T| is being halved at each call of MBSA

Sum up: O(mlogm)

Procedure 2 MBSA(G,w,T)

```
let E be the set of edges of G;
if |E − T| > 1 then
  A ← UH(E − T);
  B ← (E − T)\A;
  F ← BUSH(G_{B∪T});
  if F is a spanning arborescence of G then
    S ← F;
    MBSA((G_{B∪T}, w, T));
  else
    MBSA(G, w, T ∪ B);
  end if
end if
```

Extension

- Tarjan and Gabow Algorithm for finding MBSA
  - Time complexity is $O(|V|\log|V| + |E|)$

Reference

[1]P.M. Camerini, The Min-Max Spanning Tree Problem And Some Ex- tensions, Information Processing Letters, Vol. 7, Num. 1, January 1978\newline

[2]H.N. Gabow, R.E. Tarjan, Algorithms for Two Bottleneck Optimization Problems, Journal Of Algorithms 9 (1988) 411-417\newline

Thank You!

Question?