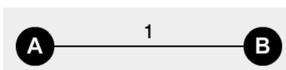
#3 Assignment - CMPT 405

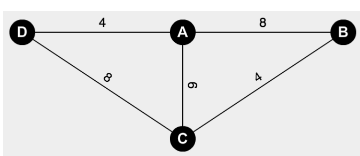
Luiz Fernando Peres de Oliveira - 301288301 - lperesde@sfu.ca

October 26, 2018

#1a)
Let G_1 be a graph with two vertices A and B and an edge (A, B) with weight 1. For every shortest path tree T_v, $v \in V$, T_v is also a MST (it is easy to see, as there is only one tree).



#1b)
Let G_2 be a graph with vertices A, B, C and D and edges $(A, B), (A, D), (A, C), (B, C)$ and (C, D), with weights $8, 4, 6, 4$ and 8, respectively. Then, no shortest path tree T_v given by Dijkstra's algorithm is a MST.



MST $= (A, C), (A, D), (B, C)$
$T_a = (A, B), (A, C), (A, D)$
$T_b = (A, B), (A, D), (B, C)$
$T_c = (A, C), (B, C), (C, D)$
$T_d = (A, B), (A, D), (C, D)$

#2)
#3)
#4)
Let T be the unrooted tree decomposition of G and T' be a nice tree decomposition of T. The idea of the algorithm is to make any node of T (preferably a internal node with many edges or minimum bag width) its root. So to ease the algorithm, we preprocess the input T: after we root T, we go through all the bag leaves b in T and create a new bag for every $v_b \in (b - parent(b))$ and make b their parent. We then run a postorder traversal on T and apply the following rules:

- **Case 1.** If bag b is a leaf in T:

If $b \neq parent(b)$, it must have come from the preprocessing of the original bag $b' - parent(b')$, and therefore $|b| = 1 \leq |parent(b)|$, meaning that we need to add introduce nodes to our nice tree decomposition T' by adding some vertices $v_{parent} \in parent(b)$ and stop when they have the same elements, so we can join the bags later; otherwise, if b and $parent(b)$ have the same elements, we are done.

- **Case 2.** If bag b is an internal node in T:

We know that if b is an internal node, then b is a parent of at least one bag b'. Then, the first step is to add a join node in T' for b and every b', where $parent(b') = b$. Also, because b is an internal node, b has a parent. We need first to get rid of all nodes in b that are not elements of $parent(b)$ (by adding forget nodes to T') and finally add some vertices $v_{parent} \in parent(b)$ and stop when b and $parent(b)$ have the same elements, so we can join the bags later (by adding introduce nodes to T').

- **Case 3.** If bag b is the root in T:

Finally, if b is the root, we only need to create a join node of all bags $children(b)$ in our final nice tree decomposition T'.

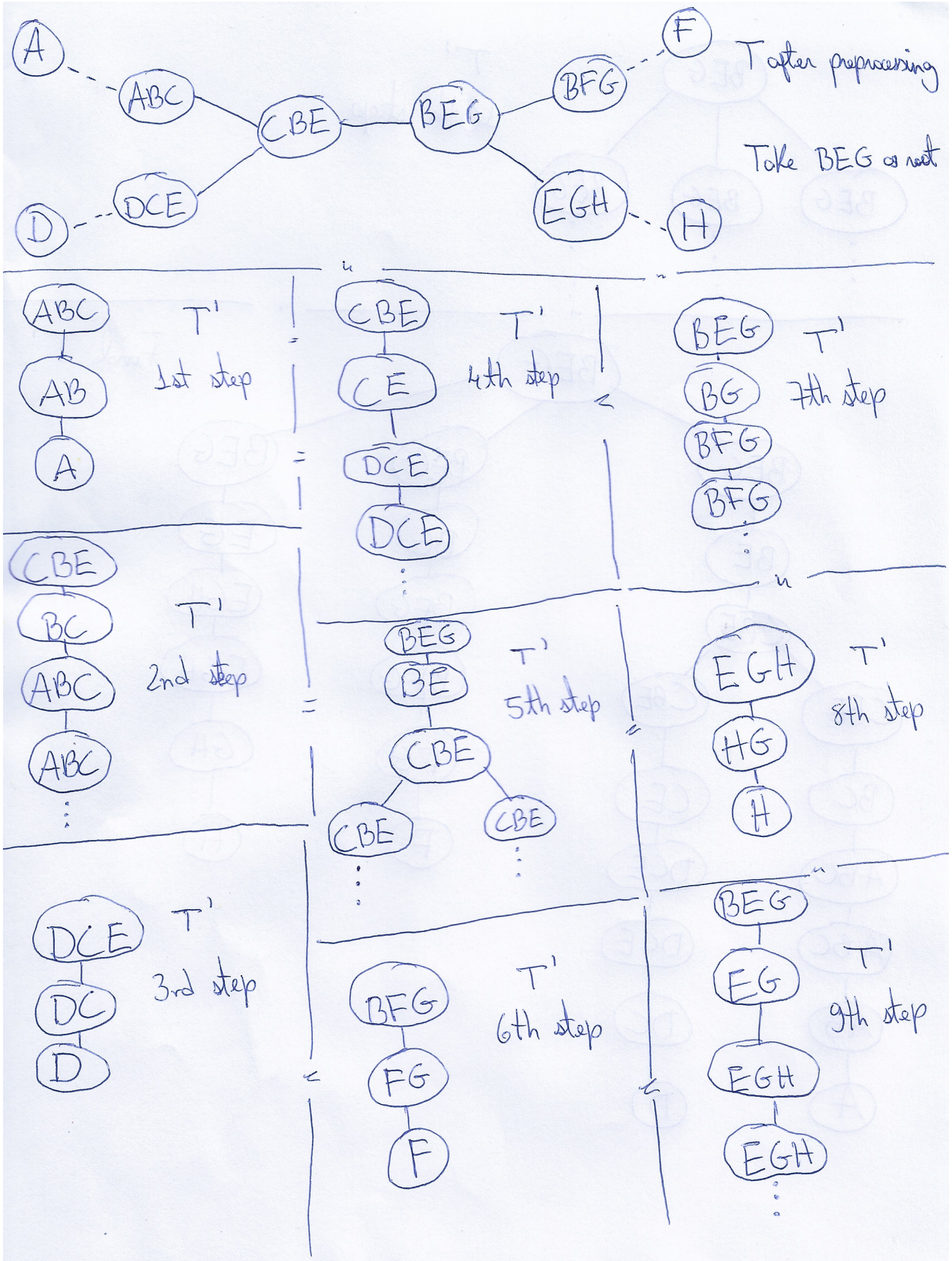
The algorithm runs in $O(nk)$ (as per the pseudocode and demonstration below)

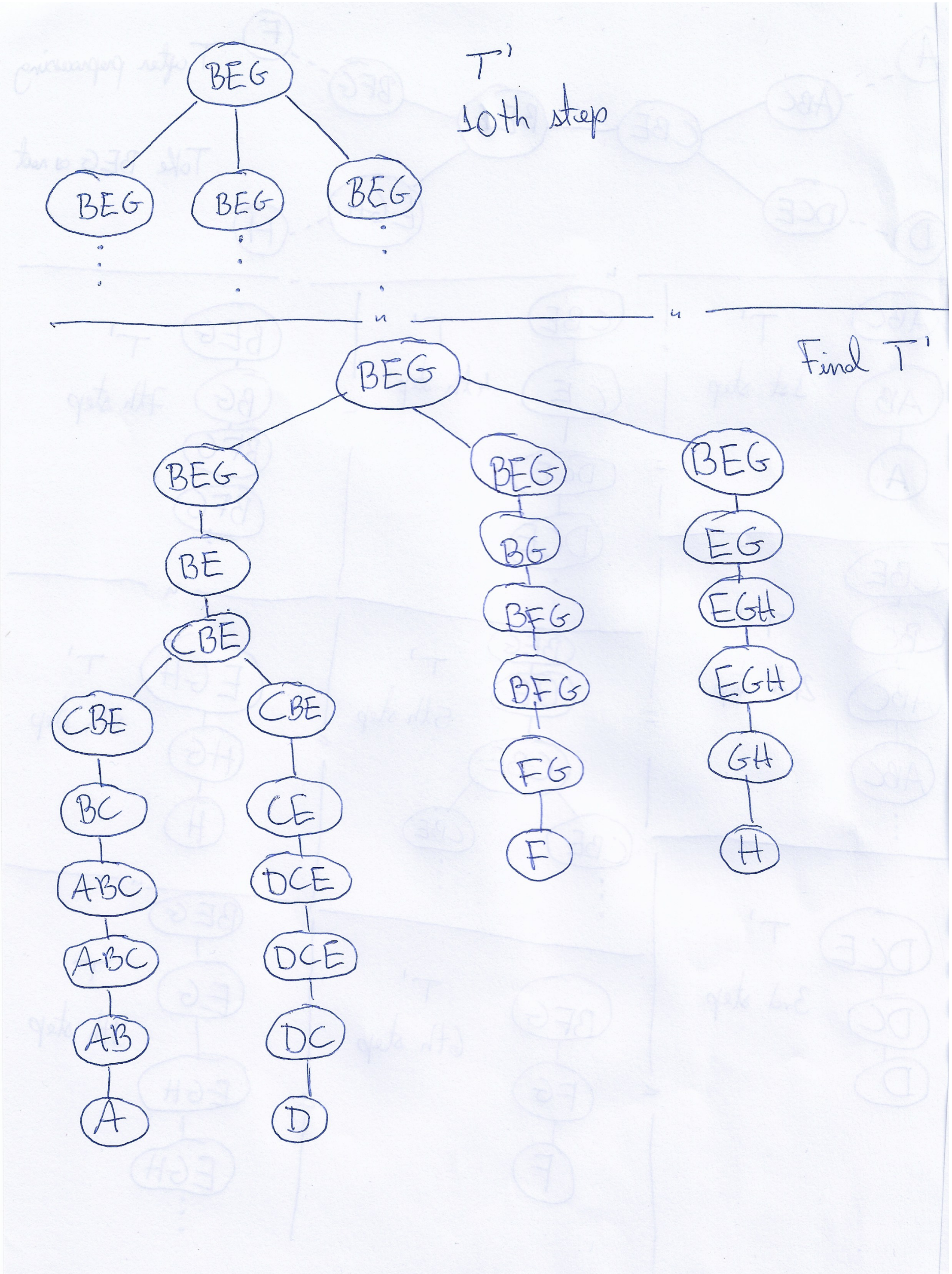
Algorithm:
Input: T

Make any internal bag (or the bag with minimum width) of T its root.
for each bag $b \in T, children(b) = \emptyset$ **do** *// all leaves*
  $free_{vs} \leftarrow b - parent(b)$
  for each $v \in free_{vs}$ **do**
    $T \leftarrow T \cup \{v\}$ such that $parent(v) = b$
  end for
end for
$T' \leftarrow \emptyset$
for each bag $b \in T_{postorder}$ **do**
  $b_{aux} \leftarrow b$
  if $children(b) = \emptyset$ **then** *// if leaf*
    $T' \leftarrow T' \cup b_{aux}$ *// add leaf node*
    while $b_{aux} \neq parent(b)$ **do**
      add introduce node $b_{aux} \cup \{v_{parent}\}$ in T', for any $v_{parent} \in parent(b)$, such that $b_{aux} \cup \{v_{parent}\} = parent(b_{aux})$
    end while
  else
    Create join node in T'
    if $parent(b) \neq \emptyset$ **then** *// if not root*
      while $\exists v \in b$, such that $v \notin parent(b)$ **do**

add forget node $b_{aux} - \{v\}$ in T', for any $v \notin parent(b)$,
such that $b_{aux} - \{v\} = parent(b_{aux})$
end while
while $b_{aux} \neq parent(b)$ **do**
add introduce node $b_{aux} \cup \{v_{parent}\}$ in T',
for any $v_{parent} \in parent(b)$, such that $b_{aux} \cup \{v_{parent}\} = parent(b_{aux})$
end while
end if
end if

end for
return T'


T after preprocessing
Take BEG as root
T' 1st step
T' 2nd step
T' 3rd step
T' 4th step
T' 5th step
T' 6th step
T' 7th step
T' 8th step
T' 9th step
A
ABC
CBE
BEG
BFG
F
D
DCE
EGH
H
AB
BC
DC
CE
BE
FG
BG
HG
EG


T'
10th step
BEG
BEG BEG BEG
Find T'
BEG
BEG
BE
CBE
CBE
BC
ABC
ABC
AB
A
CBE
CE
DCE
DCE
DC
D
BEG
BG
BFG
BFG
FG
F
BEG
EG
EGH
EGH
GH
H

#5)

References