Extended Introduction to Computer Science CS1001.py

Lecture 21: Constructing Huffman Tree: An example

Instructors: Benny Chor, Amir Rubinstein

Teaching Assistants: Michal Kleinbort, Amir Gilad

School of Computer Science
Tel-Aviv University
Spring Semester, 2017
http://tau-cs1001-py.wikidot.com

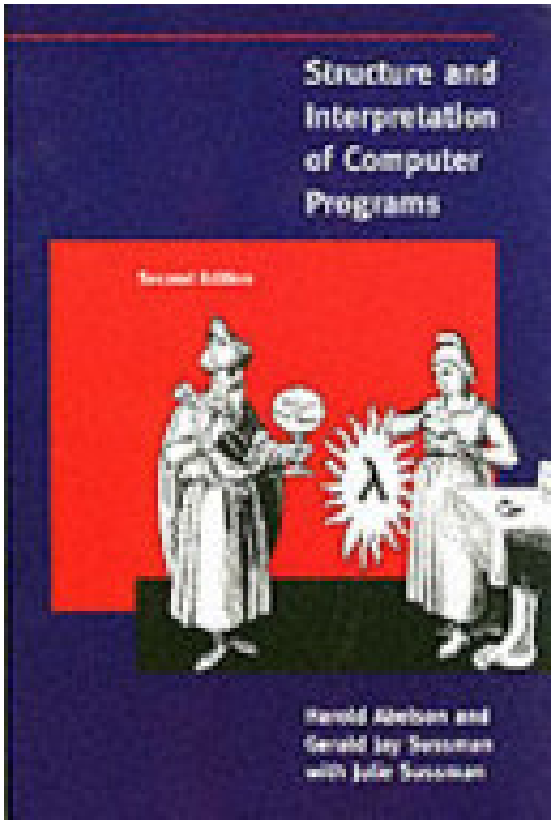
Constructing Huffman Tree

Assume that the character count is

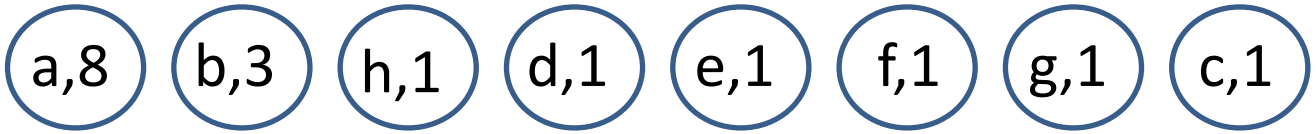
[('a', 8), ('b', 3), ('h', 1), ('d', 1),

('e', 1), ('f', 1), ('g', 1), ('c', 1)]

The example is adapted from
Structure and Interpretation of
Computer Programs,
by Harold Abelson, Gerald Jay Sussman



Priority queue: Initial value

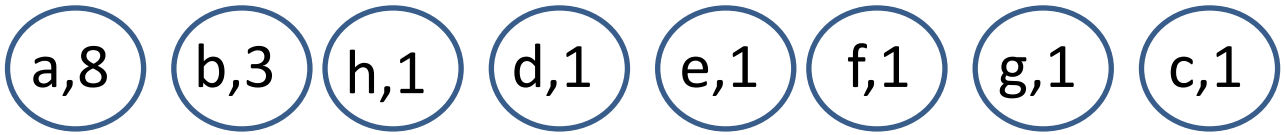


The order in the list is not important.

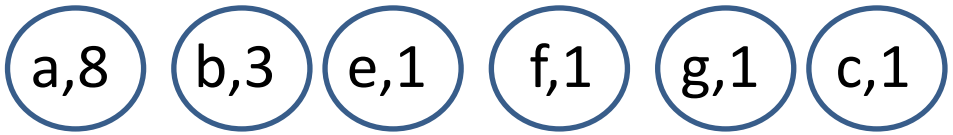
We use a naive implementation of priority queue: a list, where we insert elements at the end, and extract minimum is done by finding the minimal element and removing it.

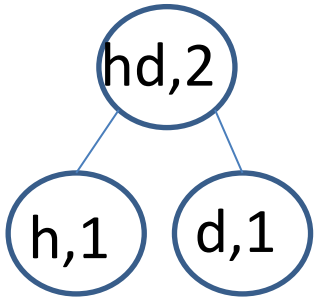
More efficient implementation is possible

Extract minimum twice:

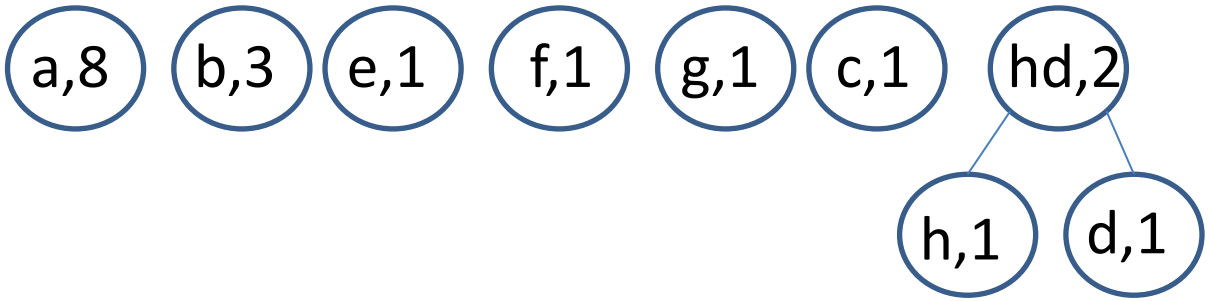


Create a new tree:

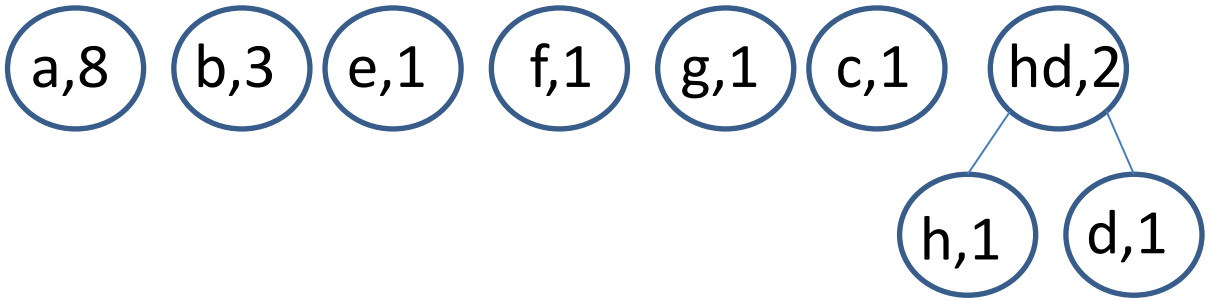


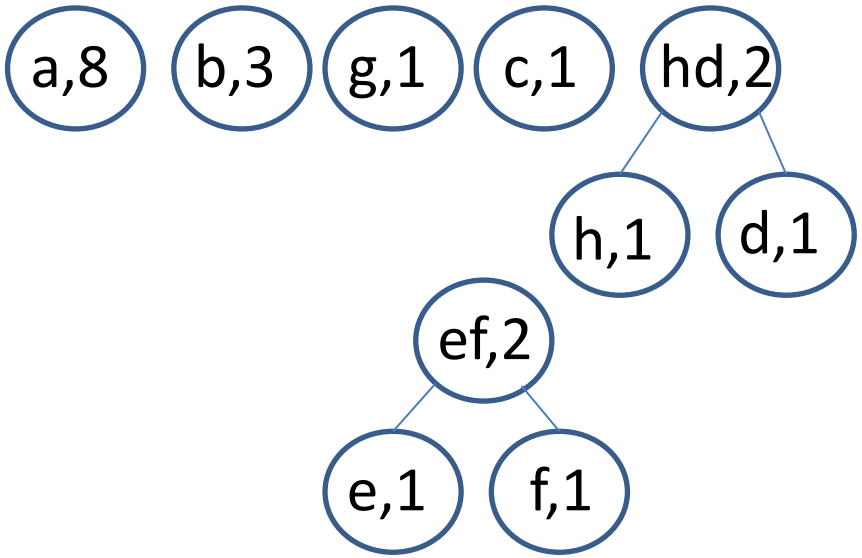


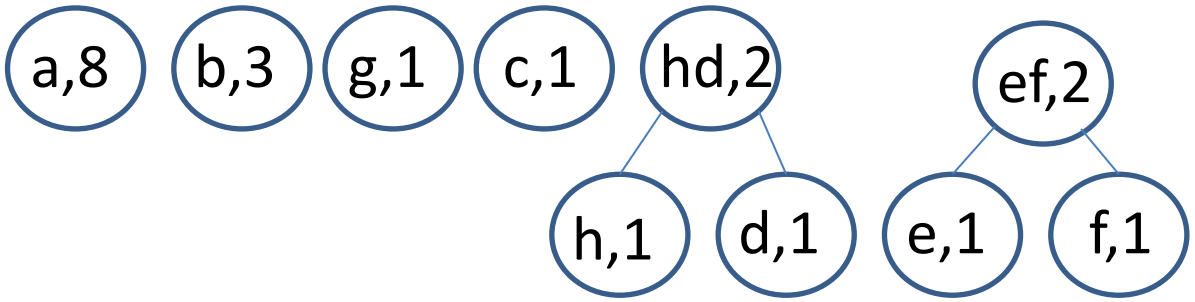
Insert the new tree into the priority queue:

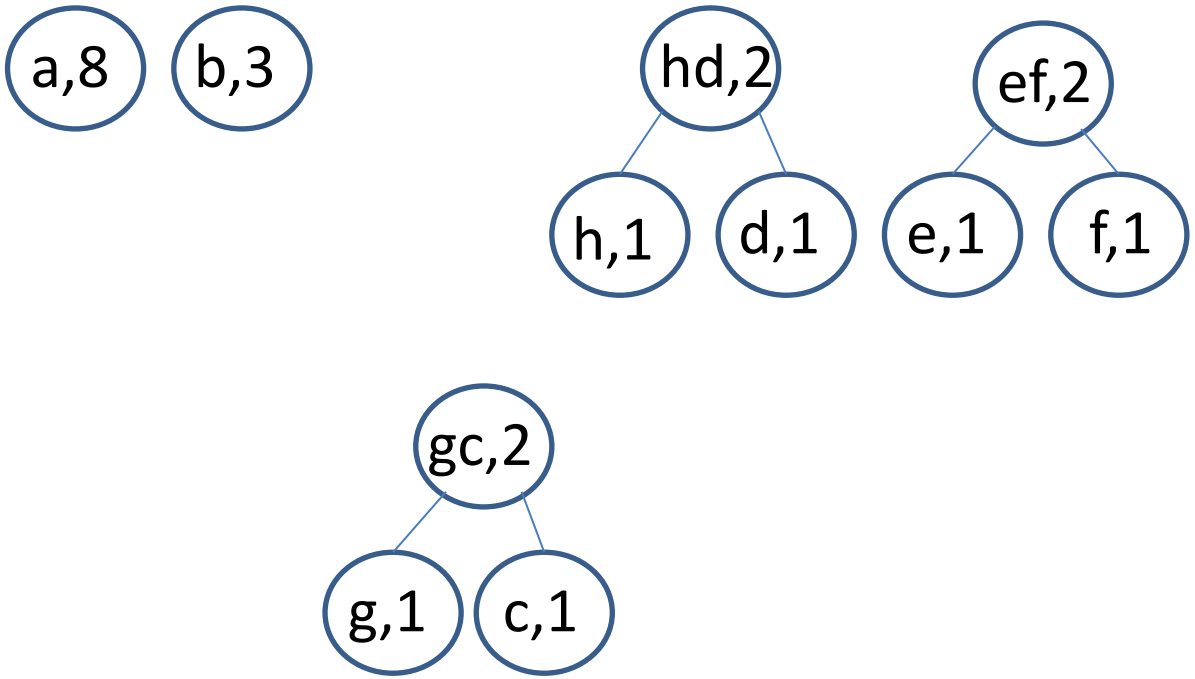


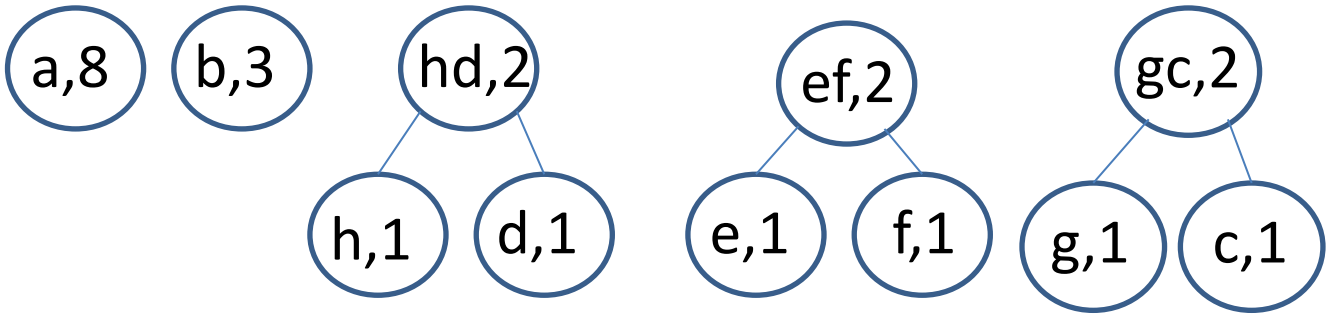
Now repeat the process:

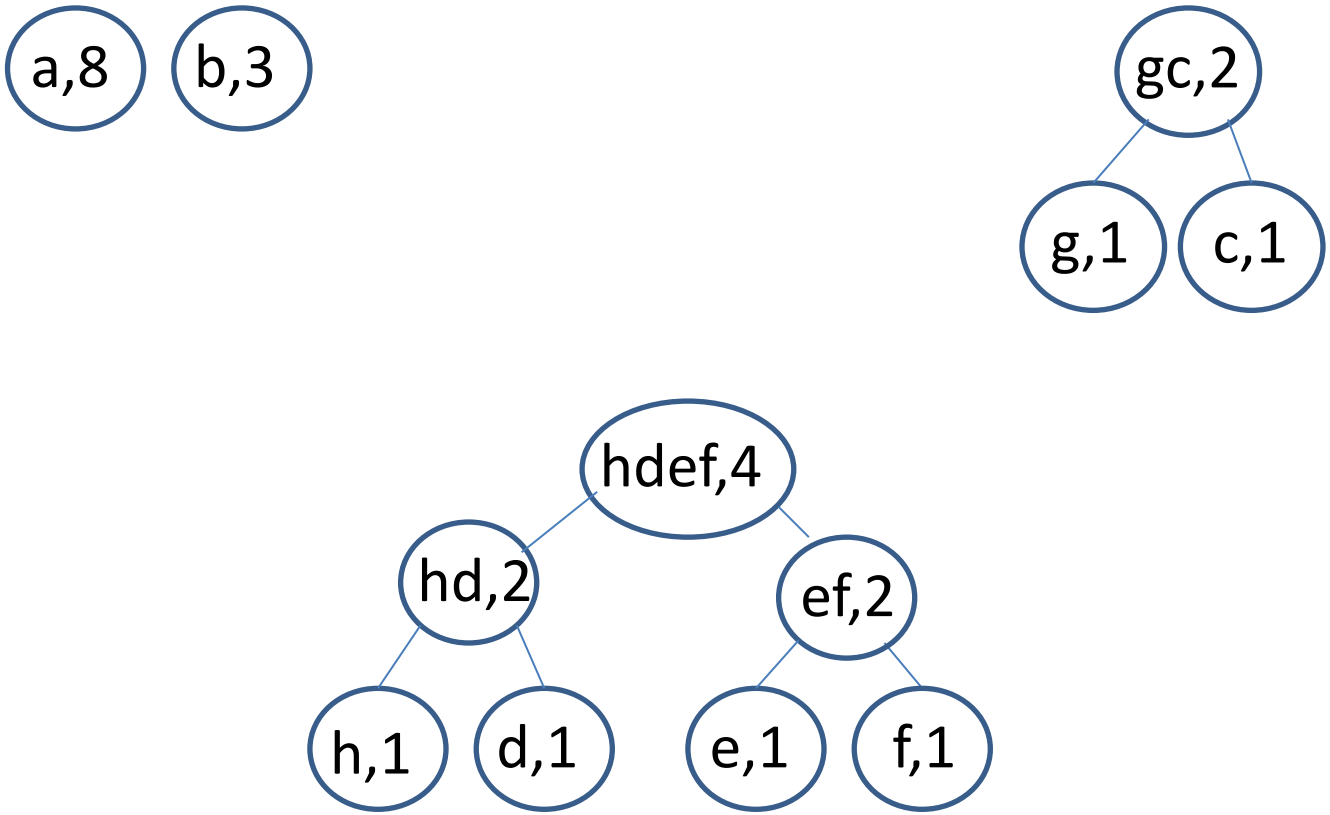


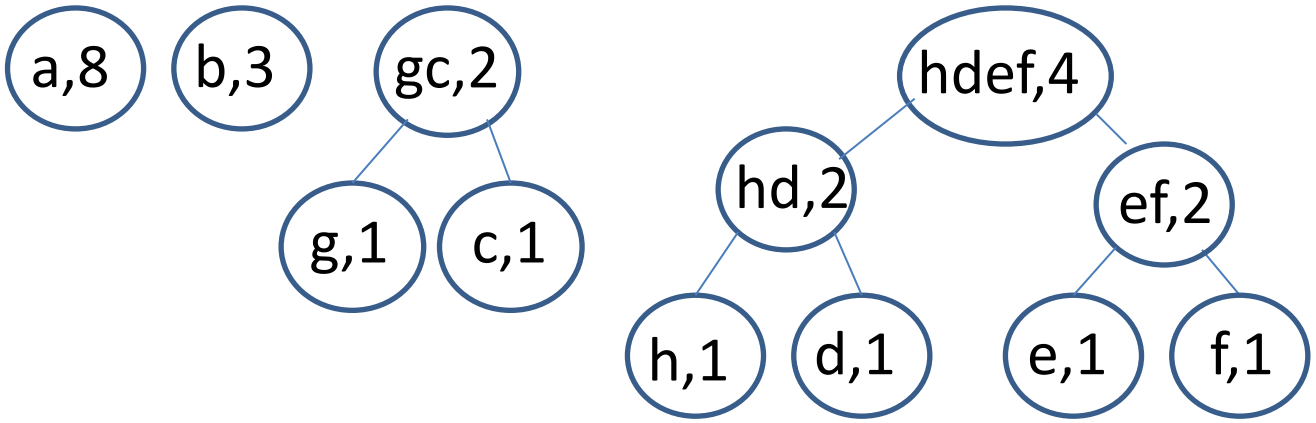

a,8
b,3
g,1
c,1
hd,2
h,1
d,1
ef,2
e,1
f,1

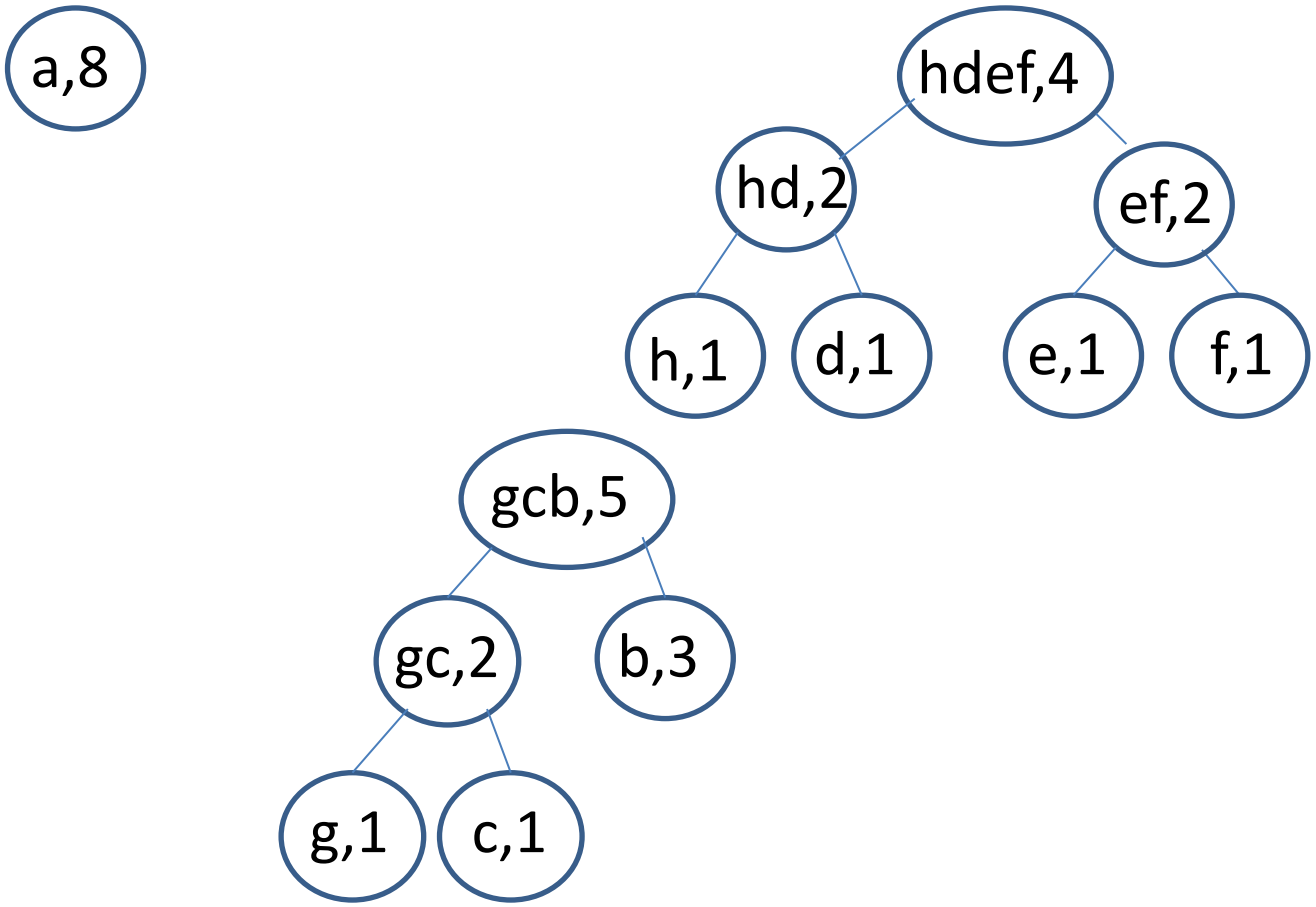

a,8
b,3
g,1
c,1
hd,2
ef,2
h,1
d,1
e,1
f,1

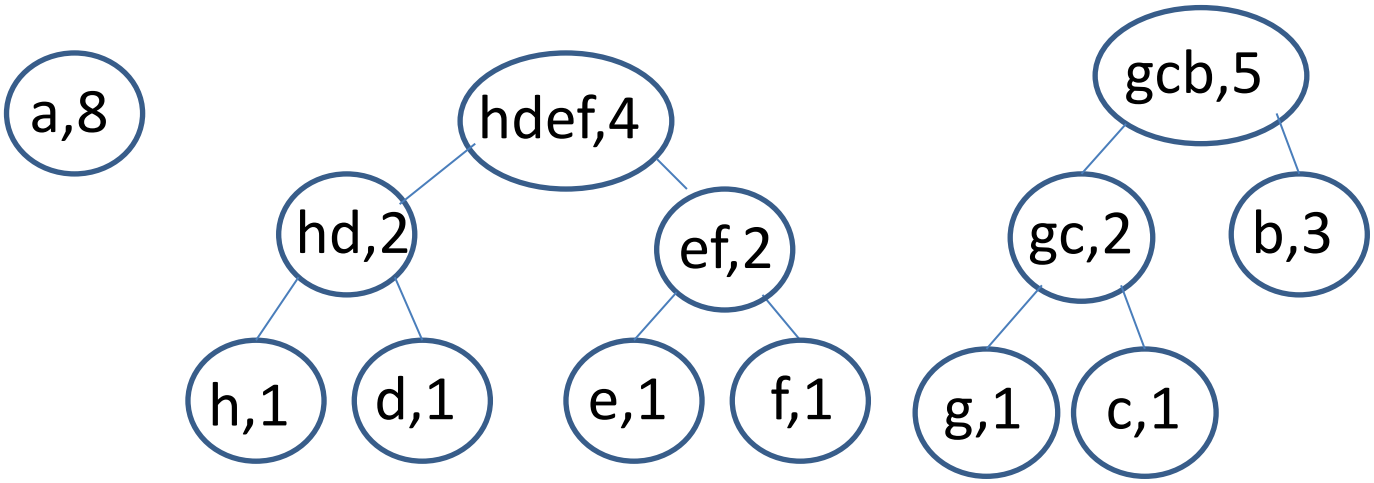

a,8
b,3
hd,2
h,1
d,1
ef,2
e,1
f,1
gc,2
g,1
c,1


a,8
b,3
hd,2
h,1
d,1
ef,2
e,1
f,1
gc,2
g,1
c,1

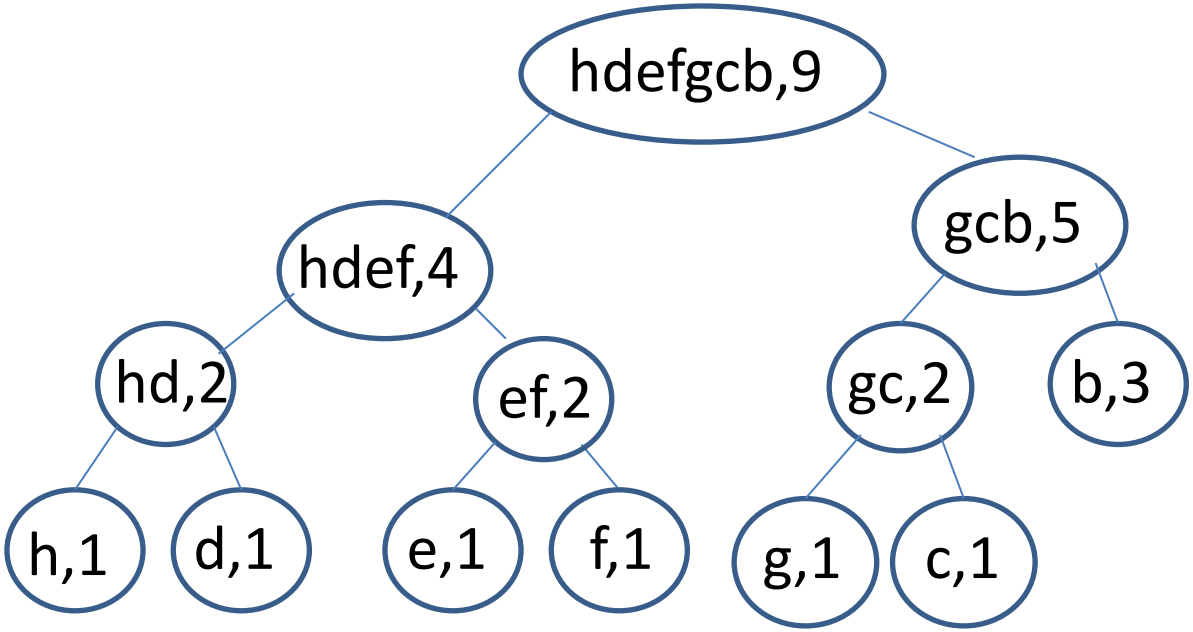

a,8
b,3
gc,2
g,1
c,1
hdef,4
hd,2
ef,2
h,1
d,1
e,1
f,1

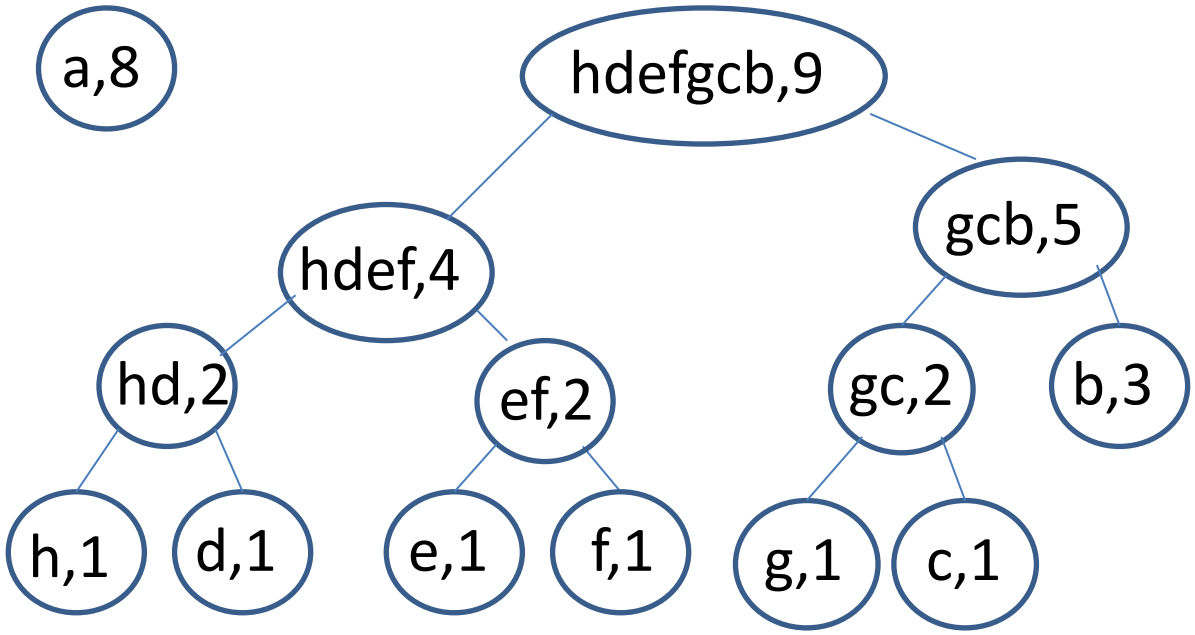

a,8
b,3
gc,2
g,1
c,1
hdef,4
hd,2
ef,2
h,1
d,1
e,1
f,1

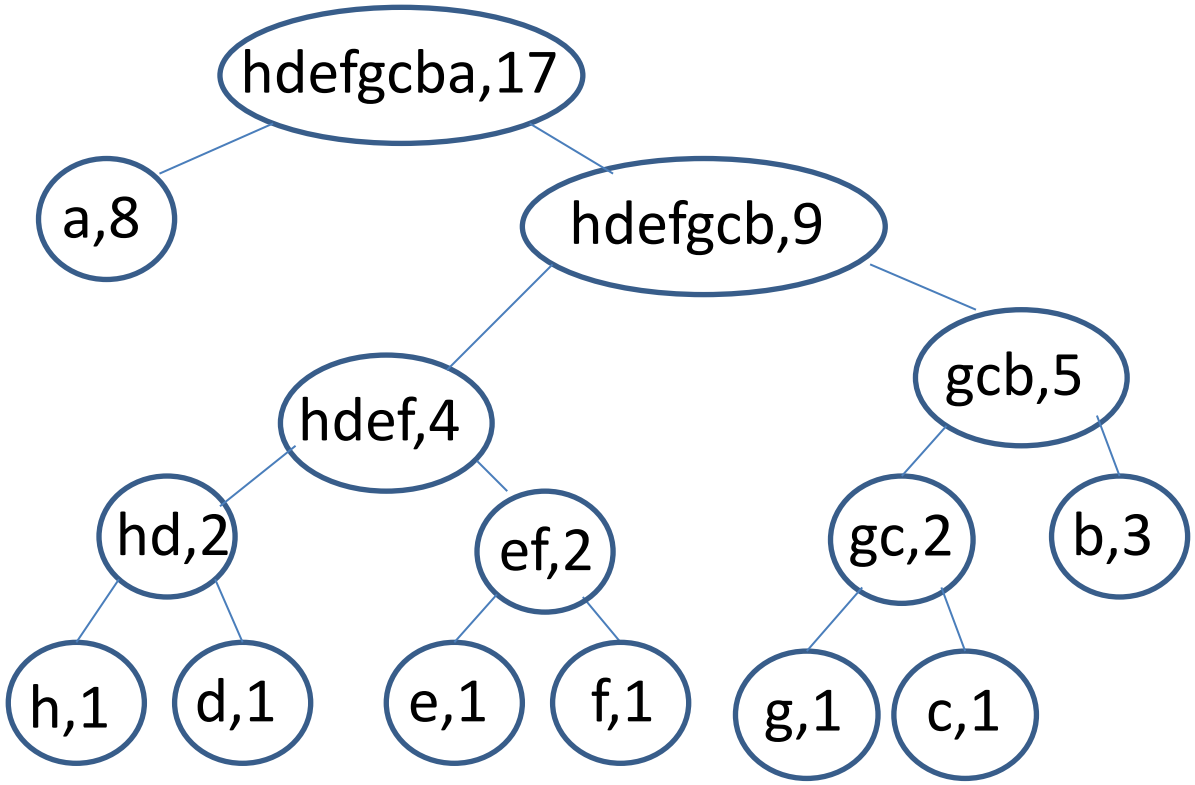

a,8
hdef,4
hd,2
ef,2
h,1
d,1
e,1
f,1
gcb,5
gc,2
b,3
g,1
c,1

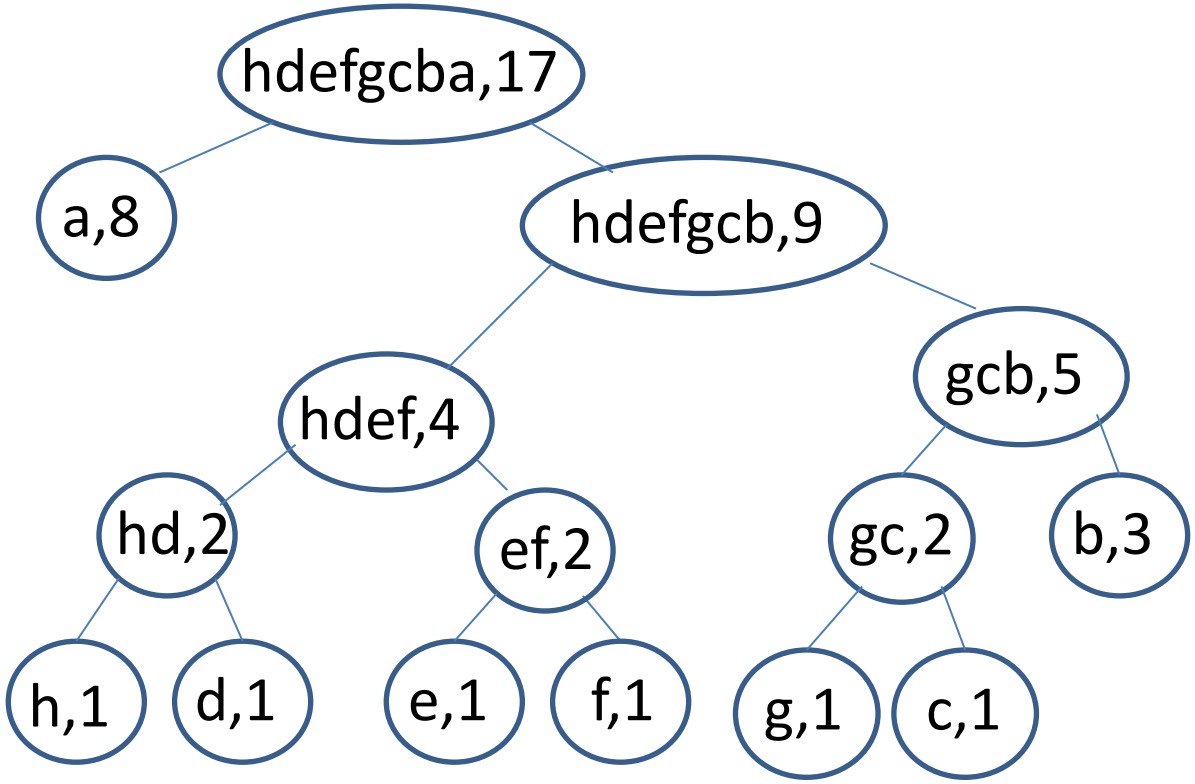

a,8
hdef,4
hd,2
ef,2
h,1
d,1
e,1
f,1
gcb,5
gc,2
b,3
g,1
c,1

a,8

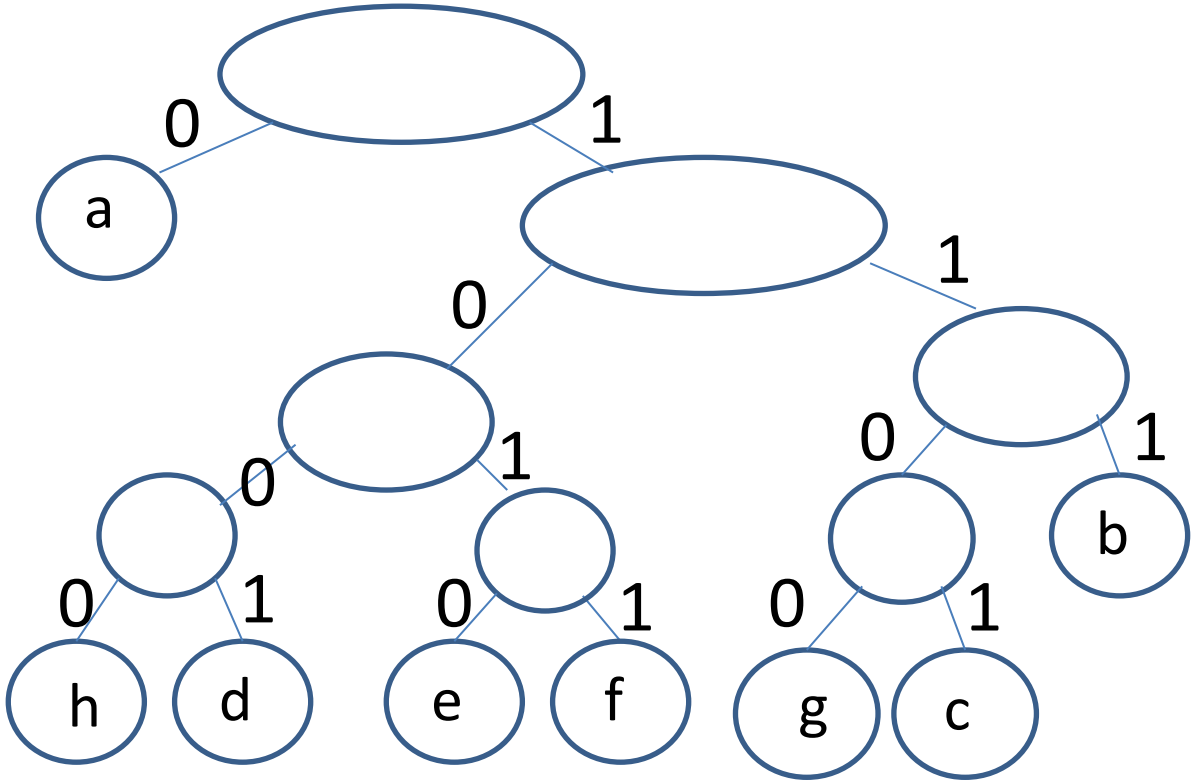



a,8
hdefgcb,9
hdef,4
gcb,5
hd,2
ef,2
gc,2
b,3
h,1
d,1
e,1
f,1
g,1
c,1


hdefgcba,17
a,8
hdefgcb,9
hdef,4
gcb,5
hd,2
ef,2
gc,2
b,3
h,1
d,1
e,1
f,1
g,1
c,1



The queue contains a single element. Loop ends



Now we can create the codes for each letter by following the paths from the root to the leaves.
Eg a → 0, h → 1000, f → 1011, ….