Huffman encoding

Q1

Suppose the letters a, b, c, d, e, f have probabilities 1/2, 1/4, 1/8, 1/16, 1/32, 1/32 respectively.

Which are the Huffman codes for the letters a, b, c, d, e, f?

The tree is as follows:

```
                 1
               /   \
              /     \
            1/2     a(1/2)
            /  \
           /    \
         1/4  b(1/4)
         /   \
        /     \
      1/8    c(1/8)
      /  \
     /    \
  1/16  d(1/16)
  /  \
e     f
```

So the codes for are:

a= 1

b= 01

c= 001

d= 0001

e= 00001

f= 00000

What is the average length of Huffman codes?

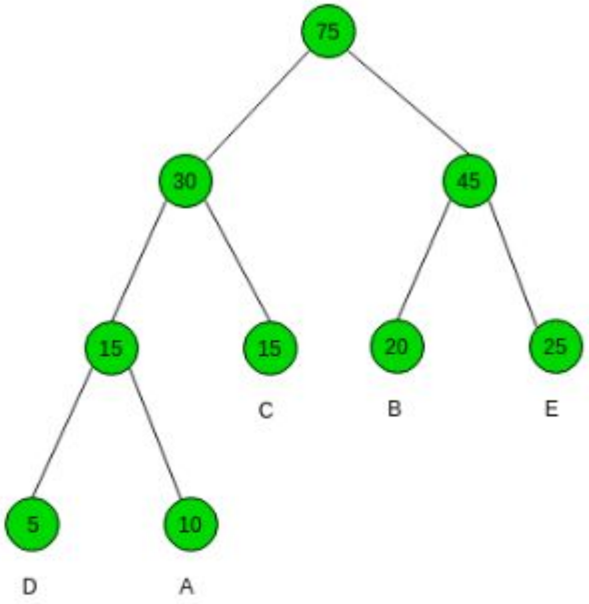
$$
\begin{aligned}
\text{Avg length} &= \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + \frac{1}{32} \times 5 + \frac{1}{32} \times 5 \\
&= \frac{16 + 16 + 12 + 8 + 5 + 5}{32} \\
&= 1.9375
\end{aligned}
$$

Q2

The minimum number of record movements required to merge five files:

A (with 10 records), B (with 20 records), C (with 15 records), D (with 5 records) and E (with 25 records) is ?

Optimal merge



Now, minimum number of record movements required = sum of internal node's value = 15 + 30 + 45 + 75 = 165

The number of comparisons of records needed?

Merge algorithm

$$(5+10-1) + (15+15-1) + (20+25-1) + (30+45-1) = 161$$

Q3

Generalize Huffman’s algorithm to ternary codewords (i.e., codewords using the symbols 0, 1, and 2)

Instead of grouping together the two with lowest frequency into pairs that have the smallest total frequency, we will group together the three with lowest frequency in order to have a final result that is a ternary tree

Activity selection/scheduling

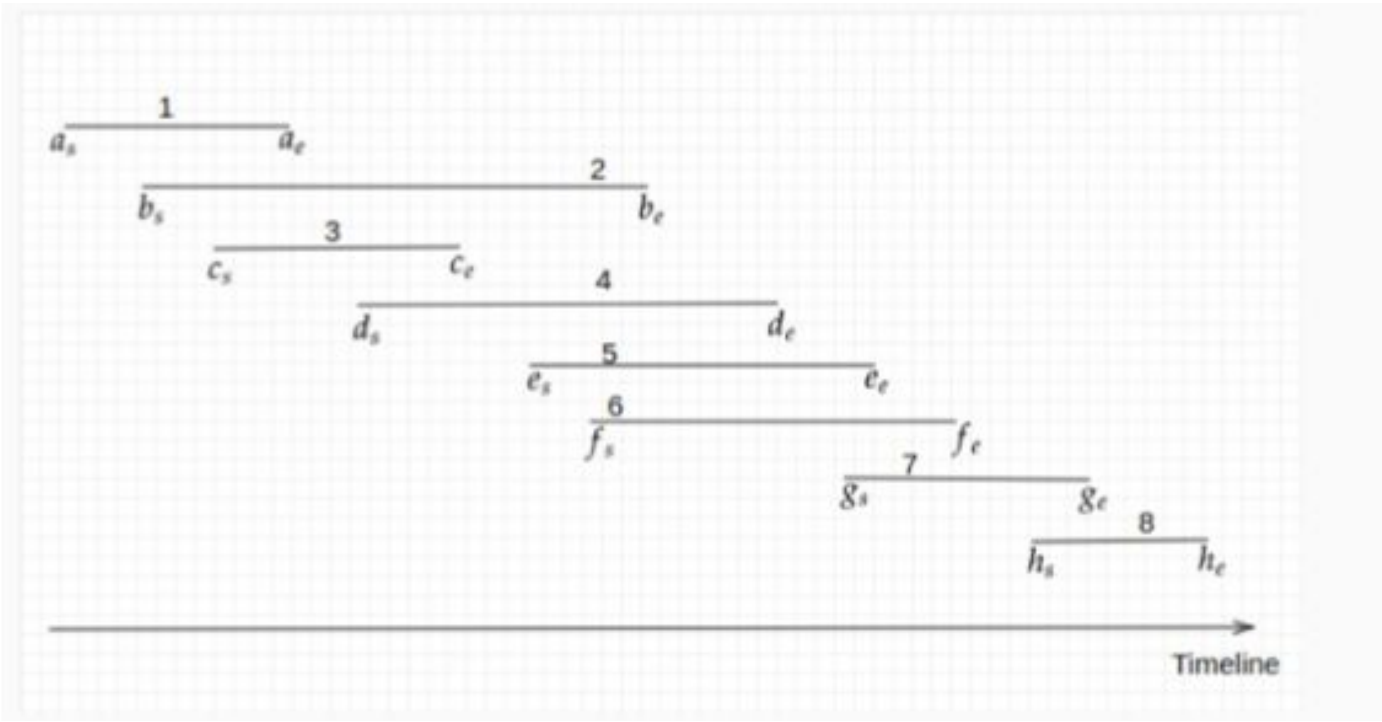
The following are the starting and ending times of activities A, B, C, D, E, F, G and H respectively in chronological order:

"a_s b_s c_s a_e d_s c_e e_s f_s b_e d_e g_s e_e f_e h_s g_e h_e "

Here, x_s denotes the starting time and x_e denotes the ending time of activity X.

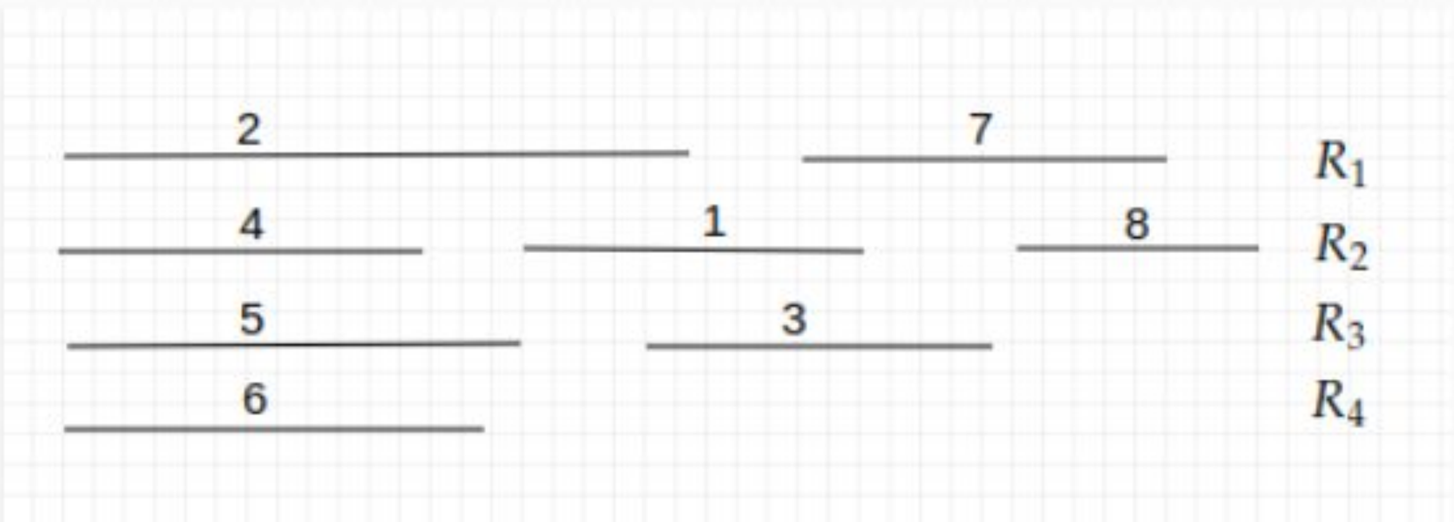
We need to schedule the activities in a set of rooms available to us. An activity can be scheduled in a room only if the room is reserved for the activity for its entire duration.

What is the minimum number of rooms required ?


1
a_s
a_e
2
b_s
b_e
3
c_s
c_e
4
d_s
d_e
5
e_s
e_e
6
f_s
f_e
7
g_s
g_e
8
h_s
h_e
Timeline

So 4 rooms needed

Rooms can be assigned as :

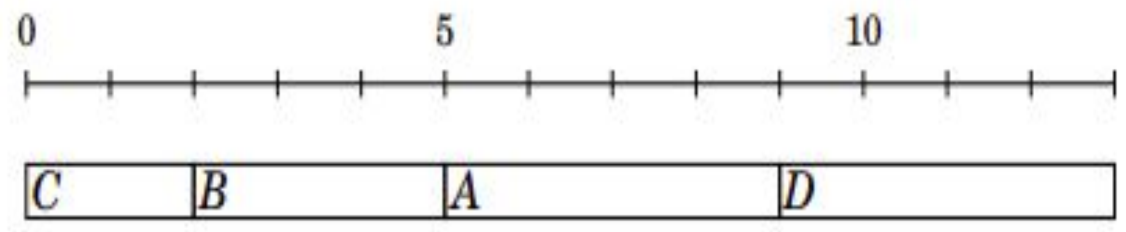


Tasks and Deadlines

Given N tasks with durations and deadlines, choose and order to perform the tasks. It is compulsory to do all the tasks. For each task, we earn (d - x) points, where d is the task's deadline and x is the moment when the task is finished. What is the maximum points we can achieve?

task	duration	deadline
A	4	2
B	3	5
C	2	7
D	4	5

In this case the optimal solution is:

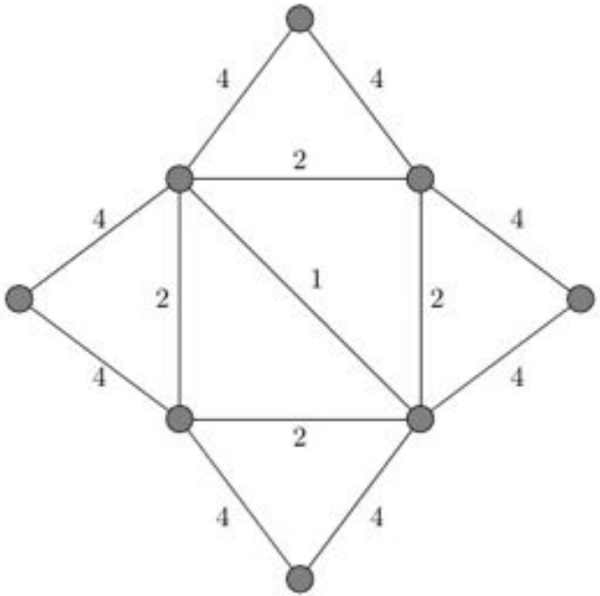


In this solution, C yields 5 points, B yields 0 points, A yields −7 points and D yields −8 points, so the total score is −10.

Surprisingly, the **optimal solution to the problem does not depend on the deadlines** at all, but a correct greedy strategy is to simply perform the tasks sorted by their durations in increasing order. The reason for this is that if we ever perform two tasks one after another such that the first task takes longer than the second task, we can obtain a better solution if we swap the tasks.

Minimum spanning tree

How many distinct minimum weight spanning trees does the following undirected, weighted graph have ?



There are 64 minimum spanning trees possible.

Two choices for edge of weight 4 for each of the outer triangle, ie, $\binom{2}{1} * \binom{2}{1} * \binom{2}{1} * \binom{2}{1} = 2^4$ and two choices for edge of weight 2 for each of the inner triangle ie, $\binom{2}{1} * \binom{2}{1} = 2^2$.

$2^4 * 2^2 = 64$

OJ

LIS: https://leetcode.com/problems/longest-increasing-subsequence/

Huffman: https://leetcode.com/problems/minimum-cost-tree-from-leaf-values/

Activity selection:

https://leetcode.com/problems/minimum-number-of-arrows-to-burst-balloons/

https://leetcode.com/problems/minimum-number-of-taps-to-open-to-water-a-garden/

Kruskal’s MST algo

Homeworks(not graded).. Because you most probably have already done these

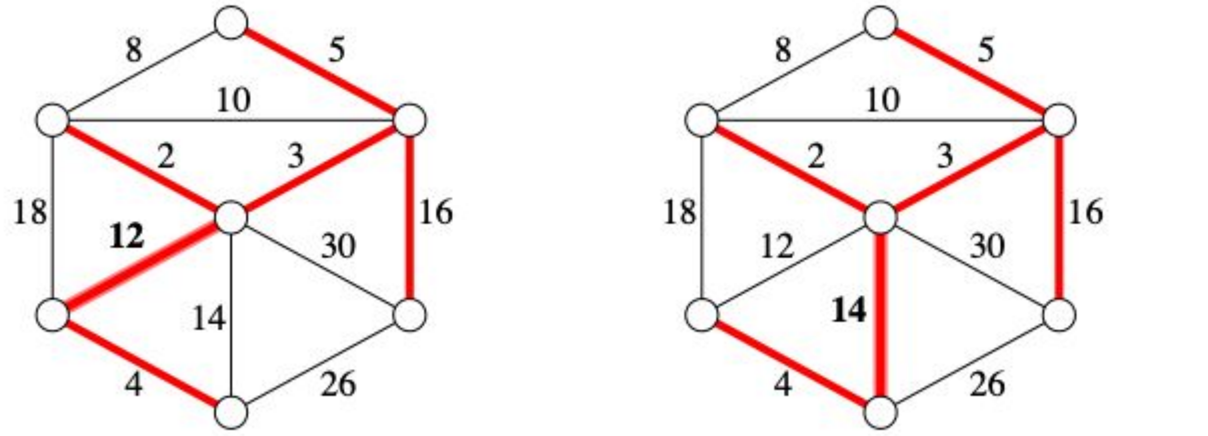
Already done in class. .

Most classical minimum spanning tree algorithms use the notions of "safe" and "useless" edges described in lecture and Jeff Erickson's lecture notes, but there is an alternative formulation. Let G be a weighted undirected graph where the edge weights are distinct. We say that an edge e is ***dangerous*** if it is the longest edge in some cycle in G and ***useful*** if it does not lie in any cycle in G.

(a) Prove that the minimum spanning tree of G contains every useful edge. *[Hint: Spanning trees are connected subgraphs containing every vertex.]*

(b) Prove that the minimum spanning tree of G does not contain any dangerous edge. *[Hint: Give an exchange argument. How can we decrease the weight of a spanning tree containing a dangerous edge?]*

Describe and analyze and algorithm to find the *second smallest spanning tree* of a given undirected graph *G* with weighted edges, that is, the spanning tree of *G* with smallest total weight except for the minimum spanning tree. Assume you have the MST(or don't ?)



(Homework) Longest Common Subsequence

Given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example,

- LCS for input Sequences “ABCDGH” and “AEDFHR” is “ADH” of length 3.
- LCS for input Sequences “AGGTAB” and “GXTXAYB” is “GTAB” of length 4.

Generating all subsequences (brute force solution) takes exponential time.

Try to identify if solutions to subproblems lead to final solution.