Algorithm assignment 7
Nisharg Gosai

Problems

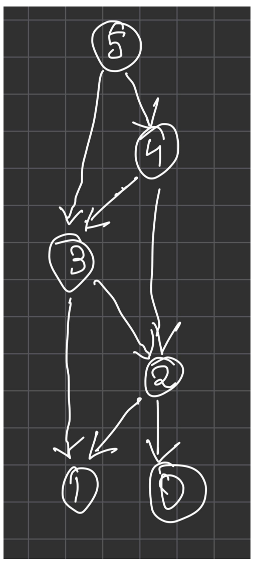
1. (20pts) Consider a modification of the rod-cutting problem in which, in addition to a price pi for each rod, each cut incurs a fixed cost of c. The revenue associated with a solution is now the sum of the prices of the pieces minus the costs of making the cuts. Give a dynamic programming algorithm in pseudocode to solve this modified problem. Explain the changes you had to make compared to the original problem in a few words.

```
MODIFIED-CUT-ROD(p, n, c)
    let r[0..n] be a new array
    r[0] = 0
    for j = 1 to n
        q = p[j]
        for i = 1 to j-1
            q = max(q, p[i] + r[j-i] - c)
        r[j] = q
    return r[n]
```

The inner for loop's body, where q= max(q, p[i] +r[j−i]−c), has to be modified. We change it to represent the fixed expense of cutting the piece, subtracted from earnings.
When i equals j, we also have to deal with the scenario where we don't make any cutbacks; in this scenario, the entire revenue is just p[j]. Hence, we alter the inner for loop to execute from i to j−1 rather than j.
We handle the case of no cuts with the assignment q=p[j]. Even in the absence of cuts, we would be subtracting c from the overall revenue if we didn't make these changes.

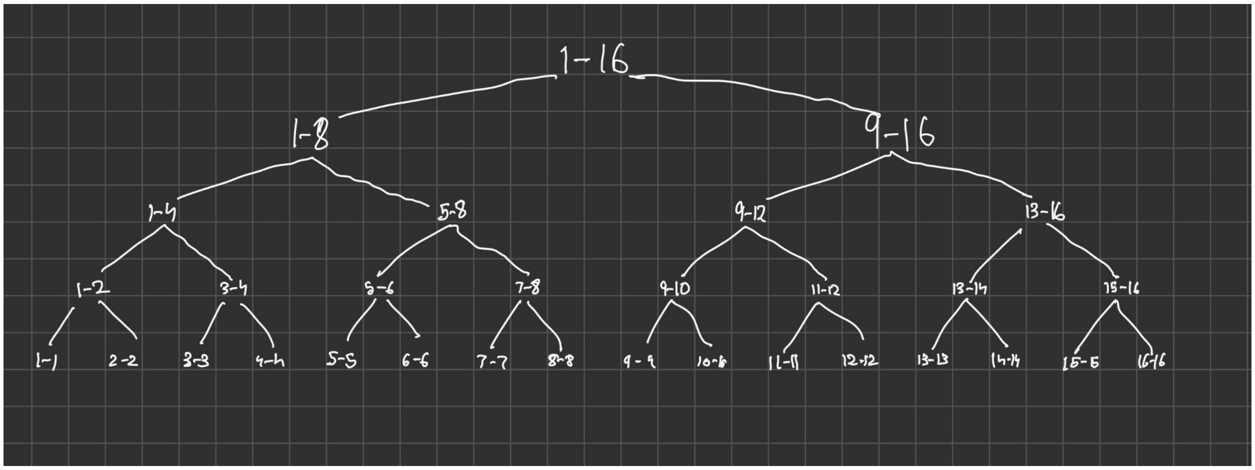
2. (20pts) Give an O(n) time dynamic programming algorithm in pseudocode to compute the nth Fibonacci number. Draw the subproblem graph. How many vertices and edges does the graph contain?

```
FIBO(n)
    If n<=1 return n
    F[0] = 0
    F[1] = 1
    For i = 2 to n
        F[i] = F[i-1]+F[i-2]
    Return F[n]
```



The subproblem graph for n=5, contains 6 vertices and 8 edges,
For the nth Fibonacci number, the subproblem graph is a simple path graph with n+1 vertices (including F[0] to F[n]) and 2n−2 edges (each of the vertices from 2 to n has two incoming edges, one from i−1 and one from i−2).

3. (20pts) Draw the recursion tree for the MERGE-SORT procedure as found in the textbook on an array of 16 elements. Explain why memorization fails to speed up a good divide-and-conquer algorithm such as MERGE-SORT.



The merge sort algorithm does not have subproblems that repeat which means no overlapping subproblems and as we can see from the tree above all nodes of the recursion tree are distinct so we don't need to store the results of the computation as it won't be needed. Therefore memorization fails to speed up Mergesort.

4. (20pts) Give pseudocode to reconstruct an LCS from the completed c table and the original sequences X = <x1, x2, ... , xm> and Y = <y1, y2, ... , ym> in O(m + n) time, without using the b table.

```
LCS(c, X, Y, i, j)
    if c[i][j] == 0
        return
    if X[i] == Y[j]
        LCS(c, X, Y, i - 1, j - 1)
        print X[i]
    else if c[i - 1][j] > c[i][j - 1]
        LCS(c, X, Y, i - 1, j)
    else
        LCS(c, X, Y, i, j - 1)
```

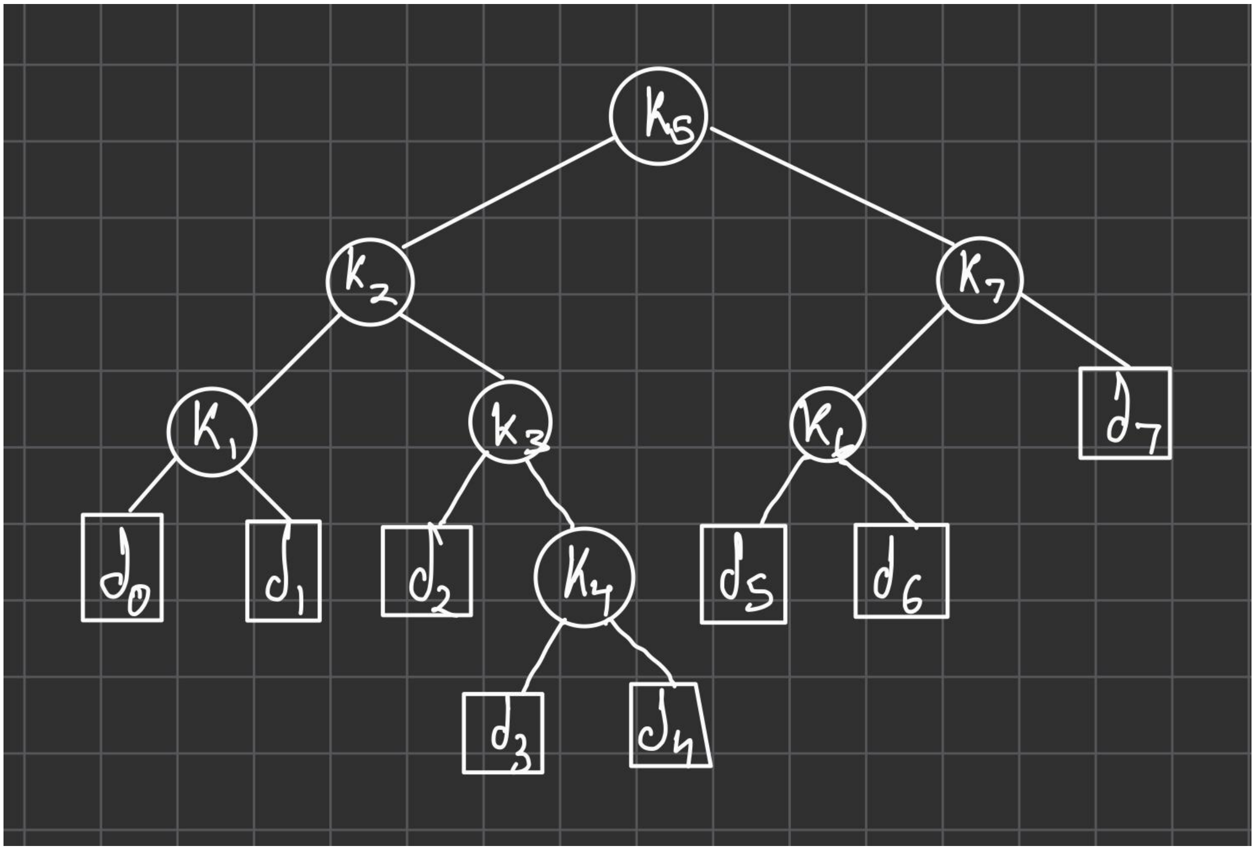
5. (30pts) Determine the cost and structure of an optimal BST for a set of n = 7 keys with the following probabilities:

i	0	1	2	3	4	5	6	7
p_i		0.04	0.06	0.08	0.02	0.10	0.12	0.14
q_i	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05

Note, that this question refers to a more general implementation of OBST that can be found in the textbook, not the one discussed in class. The qi probabilities are for dummy keys. Trace the execution of the pseudocode found in the textbook, and provide a drawing of the optimal BST, and the contents of the arrays e, and root.

e	1	2	3	4	5	6	7	8
7	3.12	2.61	2.13	1.55	1.20	0.78	0.34	0.05
6	2.44	1.96	1.48	1.01	0.72	0.32	0.05	
5	1.83	1.41	1.04	0.57	0.30	0.05		
4	1.34	0.93	0.57	0.24	0.05			
3	1.02	0.68	0.32	0.06				
2	0.62	0.30	0.06					
1	0.28	0.06						
0	0.06							

root	1	2	3	4	5	6	7
7	5	5	5	6	6	7	7
6	3	5	5	5	6	6	
5	3	3	4	5	5		
4	2	3	3	4			
3	2	3	3				
2	2	2					
1	1						



The minimum expected cost for OBST is 3.12