Summer 2021-2022 | CSE 4810 | Algorithm Engineering | Lab Final Assignment

Mark distribution: 5 + 5 + 5 + 5 = 20

1. Sort the following functions in terms of their complexity.

a) $f_1(n) = n^{2.01} \log(n^{2019})$

b) $f_2(n) = log(n)$

c) $f_3(n) = 1.00001^n$

d) $f_4(n) = n^2 log(n^2)$

e) $f_5(n) = \sum_{i=1}^{n} i$

2. Consider the following code:

```
def function1(array, lo, hi):
    length = hi - lo
    if length == 1:
        return 0
    count = 0
    for i in range(length):
        for j in range(length):
            count += 1
    mid = length // 2 + lo
    function1(array, lo, mid)
    function1(array, mid, hi)
    return count
```

i. Draw a recursion tree for the function above.

ii. What would be the height of the tree?

iii. What would be the complexity of the function?

3.

a) In the lab, we tried to figure out whether it is possible or not (binary) to make a sum given unlimited coins of several types. Let's change the coin change problem to increase the complexity a bit more. You now want to count the number of ways a specific 'total value' can be generated given a set of coin values.

A1, A2, . . ., An. You can use each coin as many times as you want. For example, given coins. 1, 2, 5, if you want to construct a total of 5, 3 possible ways to accomplish this would be:

- (1, 1, 1, 2)
- (1, 2, 2)
- (5)

i. Analyze this new formulation of the problem to find optimal substructure (show that the problem can be defined in terms of sub-problems, and those sub-problems can have sub sub-problems, and often these sub-sub-problems overlap.
ii. What would the DP table look like? What would the row and column correspond to? For coins of values 1, 2, 3, 4, and a total of 9, construct the DP table.
iii. What is the asymptotic runtime of this approach? What about the space this algorithm consumes? Can you find the big-O of the space required to store the DP table in terms of the input size (number of coins and total)?
iv. Now, assume the complexity of the problem is increased further by limiting how many times. You can use each coin. Assume this limit is k. How will you adapt your algorithm? What will the DP table look like this time?
v. Again, imagine the limit on coin usage is unique to that coin. How will you adapt your engineered solution this time? For example: Coins of values 1, 2, 3, 4 are available. The limits of usage of each of these coins in order are 8, 4, 2, 1. Imagine you need to construct a total of 20. For this specific example, create and populate a DP table and highlight the path to the answer.

b) Yet another variant of this problem can be formulated like this: Given n types of coins of values A1, A2. . ., An and maximum usage limit of these coins C1, C2, . . ., Cn respectively, you have to find the number of different values (from 1 to m), which can be produced using these coins. Please outline your solution, identify the optimal sub-structure, analyze how the DP table might. look like and the run-time and space complexities.

4. Find a valid topological sorting for the following graph using DFS.