CSCI 570 Homework 1

Due Date: Sept. 07, 2023 at 11:59 P.M.

1. Arrange these functions under the Big-$\mathcal{O}$ notation in increasing order of growth rate with $g(n)$ following $f(n)$ in your list if and only if $f(n) = \mathcal{O}(g(n))$ (here, $\log(x)$ is the **natural logarithm**[1] of x, with the base being the Euler's number e) :
$$2^{\log(n)}, 2^{3n}, 3^{2n}, n^{n\log(n)}, \log(n), n\log(n^2), n^{n^2}, \log(n!), \log(\log(n^n)).$$

2. Show by induction that for any positive integer k, $(k^3 + 5k)$ is divisible by 6.

3. Show that $1^3 + 2^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{4}$ for every positive integer n using induction.

4. Consider the following prime filtering algorithm that outputs all the prime numbers in $2, \ldots, n$ (the pseudo code is presented in Algorithm 1).
    - Please prove this algorithm is correct (that is, a positive integer k that $2 \leq k \leq n$ is a prime if and only if $isPrime(k) = \textbf{True}$).
    - Please calculate the time complexity under the Big-$\mathcal{O}$ notation.

Algorithm 1 Prime Filtering

```
Input: a positive integer n ≥ 2
initialize the Boolean array isPrime such that isPrime(i) = True for i = 2, ..., n
for i = 2 ... n do
  for j = 2 ... ⌊n/i⌋ do
    if i × j ≤ n then
      isPrime(i × j) ← False
    end if
  end for
end for
```

5. Amy usually walks from Amy's house ("H") to SGM ("S") for CSCI 570. On her way, there are six crossings named from A to F. After taking the first course, Amy denotes the six crossings, the house, and SGM as 8 nodes, and write down the roads together with their time costs (in minutes) in Figure 1. Could you find the shortest path from Amy's house to SGM? You need to calculate the shortest length, and write down all the valid paths.

[1] https://en.wikipedia.org/wiki/Natural_logarithm

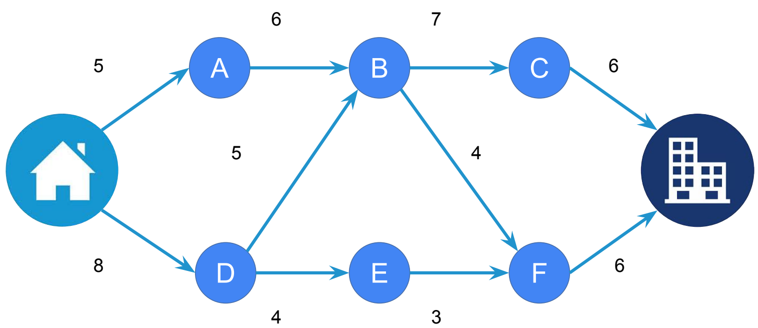

Figure 1: Problem 4's Graph

6. According to the Topological Sort for DAG described in Lecture 1, please find one possible topological order of the graph in Figrue 2. In addition, could you find all the possible topological orders?

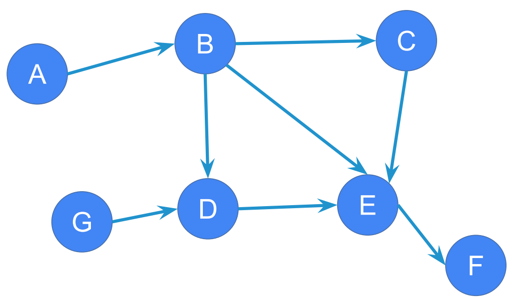

Figure 2: Problem 5's Graph

7. A binary tree[2] is a rooted tree in which each node has two children at most. A complete binary tree is a special type of binary tree where all the levels of the tree are filled completely except the lowest level nodes which are filled from as left as possible. For a complete binary tree T with k nodes, suppose we number the node from top to down, from left to right with 0, 1, 2, ..., $(k-1)$. Please solve the following two questions:

   - For any of the left most node of a layer with label t, suppose it has

[2] `https://en.wikipedia.org/wiki/Binary_tree`

at least one child, prove that its left child is $2t + 1$.

- For a node with label t and suppose it has at least one child, prove that its left child is $2t + 1$.

8. Consider a full binary tree (all nodes have zero or two children) with k nodes. Two operations are defined: 1) *removeLastNodes()*: removes nodes whose distance equals the largest distance among all nodes to the root node; 2) *addTwoNodes()*: adds two children to all leaf nodes. The cost of either adding or removing one node is 1. What is the time complexity of these two operations, respectively? Suppose the time complexity to obtain the list of nodes with the largest distance to the root and the list of leaf nodes is both $O(1)$.

9. Given a sequence of n operations, suppose the i-th operation cost 2^{j-1} if $i = 2^j$ for some integer j; otherwise, the cost is 1. Prove that the amortized cost per operation is $O(1)$.

10. Consider a singly linked list as a dictionary that we always insert at the beginning of the list. Now assume that you may perform n insert operations but will only perform one last lookup operation (of a random item in the list after n insert operations). What is the amortized cost per operation?