COMP S265F Design and Analysis of Algorithms
Assignment 1
Due: Apr 15 (Thu), 2021, 23:59

Please submit your answers in Python program files and a single PDF file to OLE. Show your steps clearly.

Question 1 (15 marks). Find the greatest common divisor of 265 and 380 using

(a) the Euclid's algorithm with mutual subtractions.

(b) the Euclid's algorithm with modulo operations.

Question 2 (35 marks). Given an array $X[0..n-1]$ of n numbers, the following algorithm computes all the subarray sums as a two-dimensional Y such that for any $0 \le i \le j \le n-1$, the entry $Y[i][j]$ stores the sum of the subarray $X[i..j]$:

```
for i ← 0 to n − 1 do
    for j ← i to n − 1 do
        Add all the entries X[i] to X[j]
        Store the sum to Y[i][j]
    end for
end for
```

(a) Analyze the time complexity of the above algorithm.

(b) What is the bottleneck of the above algorithm?

(c) The above algorithm is implemented in `q2.py`. Revise the Python program to improve its time complexity to $O(n^2)$.

(d) Show that the time complexity of your algorithm in (c) is $O(n^2)$.

Question 3 (25 marks). You are given a n-node complete binary tree T, where $n = 2^{k+1} - 1$ for some integer $k \ge 0$. Note that in a complete binary tree,

- all leaves have the same depth (which equals to the tree height), and
- all internal nodes have two children.

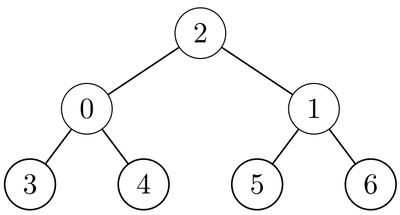
The tree height of T is k because

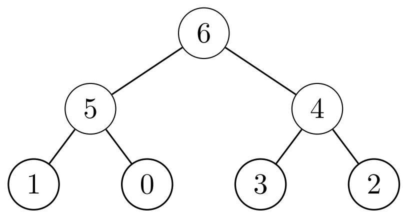
$$1 + 2 + 2^2 + \cdots + 2^k = \frac{2^{k+1} - 1}{2 - 1} = n\ .$$

Note also that $k = \log(n+1) - 1 = O(\log n)$.

Suppose that each node $x \in T$ is uniquely labeled by an integer, denoted by $\ell(x)$. We say that a node a is a *local minimum* if $\ell(a)$ is less than the labels $\ell(b)$ of all nodes b connected to node a by an edge (i.e., node b can only be the parent or child of node a). There may be more than one local minimum in T, but we are interested to find only one of them.

(a) Identify a local minimum in each of the following two complete binary trees.





(b) The program `q3.py` contains a function `traverse(node)` for traversing all the nodes in the subtree rooted at `node`. Revise this function to find one local minimum in the subtree rooted at node `node`. Your function should run in $O(\log n)$ time, where n is the number of nodes in the subtree rooted at `node`.

(c) Use the Master Theorem to show that your function in (b) has a time complexity of $O(\log n)$.

Question 4 (25 marks). You are closing your bank account at a branch of the Open Bank of Hong Kong. The bank is going to pay you the account balance in cash, yet there are only coins, but not banknotes, in the branch. Given a positive integer account balance $\$b$, you would like to receive the minimum number of coins that equals $\$b$, where the available coins are of value $\$1, \$2, \$5, \$10$, and there are enough supplies of all coin values for any possible coin combination.

(a) Write a program `q4.py` that reads in a user-specified positive integer b and outputs the minimal list of coins that equals $\$b$ in descending order.

Sample Input 1

```
9
```

Sample Output 1

```
[5, 2, 2]
```

Sample Input 2

```
76
```

Sample Output 2

```
[10, 10, 10, 10, 10, 10, 10, 5, 1]
```

(b) Prove that your algorithm in (a) is correct.