MTH407:Algorithms and Complexity

End-Semester Examination on 27th April 2025

Instructions

1. You have 3 hours (09:15 — 12:15) to complete this examination
2. Each question carries 5 marks.
3. Give *brief* explanations for your answers where required.
4. When asked to give an algorithm, either provide pseudo-code, or a program in some standard programming language (state which language you are using!).
5. When asked to calculate a function, give its value in mathematical terms as precisely as possible.
6. When asked to estimate a function, give its asymptotic growth in terms of standard notations like Big-O and little-o of standard functions like powers, exponentials, logs and their products.

Question 1

Consider the following Python program:

```
def divfn(n):
    d1 = 1
    d2 = n
    ans = 0
    while d1 < d2:
        if d1*d2 == n:
            ans += d1+d2
        d1 += 1
        d2 = n//d1
    if d1*d2==n and d1==d2:
        ans += d1
    return ans
```

a. Calculate the function $M(n)$ that counts the number of multiplications in this algorithm as a function of the input n.

b. If the input n is specified using its digits, is the running time of this program a polynomial in the size of the input?

Question 2

Order the following functions in increasing order in terms of asymptotic behaviour as n goes to infinity.

$$\binom{n}{3}\ ;\ n!\ ;\ 2^{n^{1/2}}\ ;\ n+n\sin(n)\ ;\ \sum_{k=1}^{\infty} 1/k^{n+2}\ ;\ \sum_{k=1}^{n} 1/k$$

Question 3

Given a "black-box" random generator $r()$ that outputs 0 or 1 with probability 1/2 each. Assume that all calls to $r()$ are independent events.

a. Give an algorithm that uses $r()$ to output 0, 1 or 2 with probability 1/3 each. (Hint: Consider three equally probable events made out of calls to $r()$.)

b. Calculate the expected number of calls to $r()$ of the algorithm in (a).

Question 4

Given a list of incomes of all people in a city, we wish to find the lowest 25% of the earners. (For simplicity assume that everyone has different incomes.)

a. If one sorts the complete list in order to do this, what is the worst case running time as a function of the number of people?

b. Is there an algorithm that is more efficient than (a) to do this? If so, give a brief description.

Question 5

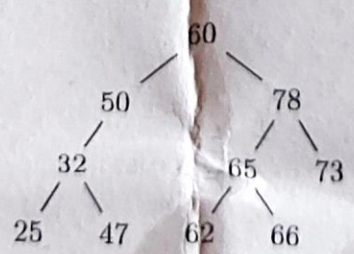
a. We are given a large database of student data including their registration number. What data structure will you use to manage this data so that we can efficiently find the data of a student associated with a certain registration number. (Assume that the student is uniquely determined by their registration number.)

b. What is the complexity (as a function of size n of the student body) of adding a student to the database?

c. A large class of m students is to be added to the database, is there a quicker way than inserting students one-by-one? Justify your answer.

Question 6

Consider the following tree:



a. Convert this into a *binary search tree* (BST) with at most one swap of two nodes.

b. Convert the resulting BST into an AVL tree with at most one rotation.

Question 7

Consider a function $g(n)$ defined recursively as follows:

$$g(0) = g(1) = g(2) = 1 \; ; \; g(n) = g(n-1) + g(n-2)^2 + g(n-3)^3 \text{ for } n \geq 3$$

a. Which algorithmic strategy will you use to design an algorithm to compute $g(n)$ efficiently using this formula?
b. Write an efficient algorithm to calculate $g(n)$.

Question 8

Consider the function recursively defined as follows:

$$f(0) = f(1) = 1 \; ; \; f(n) = 2f(n-1) + 3f(n-2) \text{ for } n \geq 2$$

a. Give an efficient algorithm to compute the value of $f(n)$ for large n.
b. Estimate the running time $T(n)$ of the algorithm as a function of the input n.