BİL 133 Combinatorics and Graph Theory

HOMEWORK 8 (42 Points)

Due Date: July 30, 2018

1 [14 Points] Get The Queen Home

Alice and Bob play the game of *Get The Queen Home* on an infinite chessboard ($\mathbf{N} \times \mathbf{N}$) with a single chess queen as follows. First Alice places the queen on any square she wants. Then Alice & Bob take turns starting with Bob to move the queen south, west, or southwest, any number of steps. The player who moves the queen to the corner of the board wins the game. We say this square at the corner is *the home square*.

Notice that if Alice places the queen carefully, then she guarantees a win. For example, if Alice places the queen to $(1,2)$, where $(0,0)$ is the home square, then she guarantees a win. We call such squares *a safe square*. Note that since the game board is symmetric, $(2,1)$ is also a safe square. Let (x_n, y_n) be the n^{th} safe square where $x_n < y_n$. First few safe squares are as following: $(1,2),(3,5),(4,7),(6,10),\ldots$

- [2 **Points**] Show that first two safe squares are $(1,2)$ and $(3,5)$.
- [4 **Points**] Prove that for each n^{th} safe square x_n and y_n differ by n, i.e., $y_n = x_n + n$.
- [4 **Points**] Prove that sequences $(x_1, x_2, \ldots)$ and $(y_1, y_2, \ldots)$ are complementary, i.e., each positive integer appears exactly once in either sequence.
- [4 **Points**] Using the theorems you proved above, devise an algorithm that returns the first n safe squares in $O(n)$ running time.

Interesting Fact: The n^{th} safe square is actually $(\lfloor n\phi \rfloor, \lfloor n\phi^2 \rfloor)$ where ϕ is the golden ratio.

2 [5 Points] More On Subset Sum

In the previous assignment, you have given an $O(n \cdot \lg n)$-algorithm for the special case of the Subset Sum problem, where $|S| = 2$. In this question, we are assuming that the array we are given is **already sorted** in ascending order and trying to reduce the time-bound to $O(n)$.
Give an $O(n)$-algorithm that takes a sorted (in ascending order) array of integers, and an integer k; and returns true if the sum of the any two distinct elements of the array is k, and returns false otherwise.

Recall that you need to prove that your algorithm terminates with the correct output in $O(n)$ time.

3 [5 Points] Even More On Subset Sum

In the previous assignment, you have given an $O(n^i)$-algorithm for the special case of the Subset Sum problem, where $|S| \leq i$. In this question, by using the observation in the above question, you are asked to show that there is an $O(n^{i-1})$-algorithm for the special case of the Subset Sum problem, where $|S| \leq i$, and $i \geq 3$.

4 [4 Points] Time-Complexity

Give an asymptotic upper-bound and an asymptotic lower-bound for the time complexity of the following algorithm, which takes an array as input.

You shall use the $O(.)$ notation for the upper-bound, and the $\Omega(.)$ notation for the lower-bound.

There should be a gap of at most $\sqrt{n}$ between your upper and lower-bounds, i.e., if you have shown that the running time of the algorithm is $\Omega(f(n))$, then your upper-bound shall be at most $O(f(n)\sqrt{n})$.

```
FancyAlgorithm(A)
    factorial = 1;
    powerofn = 1;

    for i = 1 to length[A]
        factorial = factorial . i;
        powerofn = powerofn . length[A];

        for j = 1 to powerofn/factorial
            print "Hello World"
```

(Hint: Scottish mathematician Colin Maclaurin may help you in solving this problem. If you send him an e-mail, you probably will not get a reply since he is dead. However, you may consult your Calculus book about the special case of the Taylor series named after him. In addition to that I personally find **Stirling's identity** useful, you may use it as well for your lower-bound.)

5 [4 points] More On Time Complexity

Give a tight asymptotic time-bound for the following algorithm, which takes an array as input:

```
FancyAlgorithm2(A)
    for i = 1 to length[A]
        for j = 1 to length[A]/i
            print "Hello World"
```

(Hint: Your solution should refer to Euler-Mascheroni constant. Again, do not send e-mails since both Euler and Mascheroni are dead. You may instead consult to Mascheroni's book entitled "Adnotationes ad calculum integrale Euleri" or just google for it.)

6 [5 Points] Asymptotic Comparison of Functions

You are given 5 sets of asymptotically positive functions below. For each set, rank the functions in *increasing* order of asymptotic growth rate. Notice that a set may contain functions with the same asymptotic growth rate, and if this is the case, make sure that these functions have the same rank.

- *(1 Point)* $200n^2, 5n^3 + \log n, 10n + 8/n^5$.
- *(1 Point)* $4\sqrt{n} + 4\lg(n^3), 5\log_4 n, 5\lg^2 n + 10\lg n$.
- *(1 Point)* $2^{(3\lg n)}, 500n^2, 2^{\sqrt{n}}, 5n^3$.
- *(1 Point)* $3n\lg\lg n, n\lg^6 n, 8n^2$.
- *(1 Point)* $\lg n!$, $\lg(n^n)$, $5n\lg n + 8n\lg\lg n$.

7 [5 Points] Solving Recurrences

In this question, you are given a recurrence and asked to solve it three times! Once with the substitution method, once with the recursion tree method, and once with the master method.

$$T(n) = \begin{cases} 1 & \text{if} \quad n = 1 \\ T(n) = 5T(n/2) + n^2 & \text{otherwise.} \end{cases} \tag{7.1}$$

- *(2 Points)* Solve the above recurrence by using the substitution method. Since you are to obtain a Θ-bound, you should have two separate proofs: one to establish an upper-bound, and one to establish a lower-bound.

- *(2 Points)* Solve the above recurrence by using the recursion tree method.
- *(1 Point)* Solve the above recurrence by using the master method.