ESO207 Theoretical Assignment 1 Semester II, 2016-17

Due Date: 10th February, 2017

---

1. (a) (5 points) Consider the following algorithm.

```
procedure ISPRIME($n$)

    if n==0 OR n==1 then
        return $False$

    $ARRAY\,A[n] = [1, ..., 1]$                ▷ Initialized to all ones. Index starts at 0
    $i = 2$

    while $i \leq \sqrt[2]{n}$ do
        $j = 2$
        while $i * j \leq n$ do
            $A[i * j - 1] \leftarrow 0$
            $j \leftarrow j + 1$
        $i \leftarrow i + 1$
    if $A[n-1] == 1$ then
        return $True$
    else
        return $False$
```

Find the time complexity of the above algorithm in the $O()$ notation. Show your work.

(b) (7 points) Consider the following two subroutines:

- COMBINE(A, B, C) takes 3 arrays A, B, C as input and returns an array as output.
- NEWSORT(X) takes an array X as input and returns an array as output.

```
procedure COMBINE($A, B, C$)

    $i \leftarrow 0$, $j \leftarrow 0$, $k \leftarrow 0$, $t \leftarrow 0$
    Initialize an array $Y$
    while $i + j + k < n$ do
        if $(A[i] \leq B[j])$ & $(A[i] \leq C[k])$ then
            $Y[t] \leftarrow A[i]$
            $i \leftarrow i + 1$
            if $i = \text{size}(A)$ then
                $A[i] \leftarrow \infty$
        else
            if $(B[j] < A[i])$ & $(B[j] \leq C[k])$ then
                $Y[t] \leftarrow B[j]$
                $j \leftarrow j + 1$
                if $j = \text{size}(B)$ then
                    $B[j] \leftarrow \infty$
            else
                $Y[t] \leftarrow C[k]$
                $k \leftarrow k + 1$
                if $k = \text{size}(C)$ then
                    $C[k] \leftarrow \infty$
        $t \leftarrow t + 1$
return $Y$
```

```
procedure NEWSORT($X$)

    if size($X$) > 1 then
        NEWSORT($X[0, \ldots, n/3]$)
        NEWSORT($X[n/3 + 1, \ldots, 2n/3]$)
        NEWSORT($X[2n/3 + 1, \ldots, n - 1]$)
        COMBINE($X[0, \ldots, n/3], X[n/3 + 1, \ldots, 2n/3], X[2n/3 + 1, \ldots, n - 1]$)
```

Write a recursion for the time complexity of the algorithm NEWSORT. Solve the recursion to find the time complexity of NEWSORT in the $O()$ notation.

2. (12 points) Compare and order the asymptotic growth order of the following functions in ascending order. Mention explicitly if any two functions are equivalent or if they cannot be compared. For each function $f(n)$ in the list below, what is the best function class $O(g(n))$ such that $f(n) \in O(g(n))$?

(i) $4 \times 2^{n/2}$

(ii) e^n

(iii) $500n^2 - 3n + 2$

(iv) $n!$

(v) $n^2 + n + 2000$

(vi) $n \log n$

(vii) $n \ln n$

(viii) $n \log(\log(\log n))$

(ix) $\sqrt{n}$

(x) $\binom{2n}{n}$

(xi) $\sin n$

(xii) n^n

3. (13 points) $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ are two sequences of n positive integers each such that $1 < x_i \leq y_i$ for all $1 \leq i \leq n$. Let

$$C = \sum_{i=2}^{n} |x_i - x_{i-1}|.$$

Design an $O(n)$ algorithm that given the sequence Y as input, chooses a sequence X such the value of C is maximum. Prove the correctness of your algorithm.

4. (13 points) Give an algorithm to design a tournament consisting of n players so as to correctly identify the top three players in the tournament, such that the number of matches is minimized in the tournament.

What is the minimum number of matches that such a tournament must have (as a function of n)?

Assume the following:

- for any two players, *say* A and B; if A is a better player than B, then A will defeat B in any match played between them irrespective of any factors.
- for any three players, *say* A, B and C; if A defeats (is better than) B and B defeats (is better than) C then A can be assumed to defeat (be better than) C.
- no match ever ends in a draw, i.e. between any two players we can always say one of them is better than the other.
- before the tournament no information is known about the players.