COMP108 Algorithmic Foundations

Tutorial 3 (Suggested Solution and Feedback) w/c 20th February 2017

1. (a) $O(n^4)$
   (b) $O(n^3)$
   (c) $O(n^3)$

2. To prove that $2n^2 + 3n + 4$ is $O(n^2)$, we show that

   $$\exists \text{ constants } c \text{ and } n_0 \text{ such that for any integer } n \geq n_0,\ 2n^2 + 3n + 4 \leq c\,n^2.$$

   - $2n^2 = 2n^2\ \forall n$,
   - $3n \leq 3n^2\ \forall n \geq 1$, N.B. $n \not\leq n^2$ for $0 < n < 1$
   - $4 \leq 4n^2\ \forall n \geq 1$,

   As a result, $2n^2 + 3n + 4 \leq 9n^2\ \forall n \geq 1$.
   Since 9 and 1 are constants, the function $2n^2 + 3n + 4$ is $O(n^2)$.
   N.B.: Should give reasons for the inequalities as shown above.

   Alternatively, we can prove in the following way:

   - $2n^2 = 2n^2\ \forall n$,
   - $3n \leq n^2\ \forall n \geq 3$, and
   - $4 \leq n^2\ \forall n \geq 2$,

   As a result, $2n^2 + 3n + 4 \leq 4n^2\ \forall n \geq 3$.

3. To prove that $2n^3 + 3n^2 + n\log n$ is $O(n^3)$, we show that

   $$\exists \text{ constants } c \text{ and } n_0 \text{ such that for any integer } n \geq n_0,\ 2n^3 + 3n^2 + n\log n \leq c\,n^3.$$

   - $2n^3 = 2n^3\ \forall n$,
   - $3n^2 \leq 3n^3\ \forall n \geq 1$, N.B. $n^2 \not\leq n^3$ for $0 < n < 1$ nor $n < 0$
   - $n\log n \leq n^3\ \forall n \geq 1$, $\because \log n \geq 0$ and $n^2 \leq n^3\ \forall n \geq 1$

   As a result, $2n^3 + 3n^2 + n\log n \leq 6n^3\ \forall n \geq 1$.
   Since 6 and 1 are constants, the function $2n^3 + 3n^2 + n\log n$ is $O(n^3)$.
   N.B.: Should give reasons for the inequalities as shown above.

   Alternatively, we can prove in the following way:

   - $2n^3 = 2n^3\ \forall n$,
   - $3n^2 \leq n^3\ \forall n \geq 3$, and
   - $n\log n \leq n^3\ \forall n \geq 1$,

   As a result, $2n^3 + 3n^2 + n\log n \leq 4n^3\ \forall n \geq 3$.

4. To check whether x is prime, we should check all integers from 2 to $x-1$; if x is divisible by ANY of them, then x is not prime; if x is not divisible by ALL of them, then x is prime.

   $prime = true$
   $i = 2$
   while $i \leq x-1$ and $prime == true$ **do**
   begin
       if $x\%i == 0$ **then**
           $prime = false$
       else $i = i+1$
   end
   output $prime$

   The time complexity of this algorithm is $O(x)$.

   We can speed up the algorithm by stopping earlier.

   $prime = true$
   $i = 2$
   while $i \leq \sqrt{x}$ and $prime == true$ **do**
   begin
       if $x\%i == 0$ **then**
           $prime = false$
       else $i = i+1$
   end
   output $prime$

   The time complexity of this algorithm is $O(\sqrt{x})$.

5. It is impossible to pick 10 numbers from an infinite supply of 1, 3, 7 to obtain a sum of 39. No matter which numbers we pick, every two of them have a sum being an even number. Therefore, picking 10 of them forms 5 even sums and any even number plus even number gives even number, so we cannot get a sum of 39, which is an odd number.