

Homework - Week 9

Name: Sahand Sabour

Student ID: 2020280401

1. Please upload a picture of you participating the online meeting of the course (or watching playbacks), including yourself and your devices.



2. Prove by definition: $2n^2 - 3n = O(n^2)$.

By definition, $2n^2 - 3n = O(n^2)$ means that n^2 is the upper bound for $2n^2 - 3n$.

Hence, we must find constant c and n_0 such that

$$2n^2 - 3n \leq c * n^2 \text{ for all } n \geq n_0$$

Setting $c=2$ and $n_0=0$ gives

$$-3n \leq 0 \text{ for all } n \geq 0$$

Since the condition holds for the given positive values of c and n_0 , it can be concluded that $2n^2 - 3n = O(n^2)$.

3. Let

$$P(n) = \sum_{i=0}^d a_i n^i$$

Where $a_d > 0$, be a degree- polynomial in , and let be a constant. Use the definitions of the asymptotic notations to prove:

If $k \geq d$, then $p(n) = O(n^k)$

By definition, $p(n) = O(n^k)$ means that n^k is the upper bound for $p(n)$. Hence, we must find constant c and n_0 such that

$$P(n) \leq c * n^k \text{ where } k > d \text{ and } a_d > 0 \text{ for all } n \geq n_0$$

Accordingly, for $n \geq 1$, we can have that

$$P(n) = \sum_{i=0}^d a_i n^i = a_0 n^0 + a_1 n^1 + \dots + a_d n^d \leq b_0 n^d + b_1 n^d + \dots + b_d n^d$$

Where $b_i = |a_i| \geq 0$, giving

$$P(n) \leq n^d \sum_{i=0}^d b_i \leq n^k \sum_{i=0}^d b_i$$

Setting $c = \sum_{i=0}^d b_i$ and $n_0 = 1$ satisfies the above equation and since both are positive, these positive values could be used to prove the given statement.

4. Show that the majority element problem can be reduced to the sorting problem, following the three steps of reduction.

The input to the majority element problem is an array A of n numbers.

Step 1: The same array A could be used as the input to the sorting problem.

Step 2: Using the sorting \propto convex hull problem sorts the array in ascending order and the output would be A' , sorted version of A , with lower bound of $\Omega(n \log n)$.

Step 3: Select the median of A' and check if it's the majority element via counting the number of its occurrences and comparing it to the length of array, given that majority element M appears more than half of the array: $\text{count}(M) > \text{len}(A)/2$. This would be completed in runtime with upper bound of $O(n)$.