CS301 HOMEWORK 1

Sabanci University

Mert Ziya
27800

Contents:

QUESTION 1: [20 points]

a) What is the form of the input array that triggers the worst case of the insertion sort? [5 points]

Worst case scenario for insertion sort happens when the input array is reversely sorted

b) What is the complexity of this worst–case behavior in Θ notation? [5 points]

The complexity of this worst case behavior is $\Theta(N^2)$

c) Explain how this particular form of the array results in this complexity. [10 points]

Because for each element in the array while iterating, algorithm should compare the current element with next element and switch their positions if they are not ordered(by themselves) for making the array sorted. If the array is sorted reversely, the algorithm should compare the first element $N-1$ times, then it should compare the second element $N-2$ times and so on, which results in $N-1+N-2+N-3+N-4+\cdots=\frac{N(N-1)}{2}$. As a result, there will be exactly $\frac{N^2-N}{2}$ comparisons. (there will be values for swapping operation as well, but the leading term comes from comparison operation that's why i only take comparisons to account.)

- $\frac{N^2-N}{2}=O(N^2)$ because $\frac{N^2-N}{2}\leq cN^2$ is true when $c=\frac{1}{2}$ and $N\geq n_0$ for $n_0\geq 1$.
- $\frac{N^2-N}{2}=\Omega(N^2)$ because $\frac{N^2-N}{2}\geq cN^2$ is true when $c=\frac{1}{4}$ and $N\geq n_0$ for $n_0\geq 1$.
- Since $\frac{N^2-N}{2}=O(N^2)$ and $\frac{N^2-N}{2}=\Omega(N^2)$, it means that $\frac{N^2-N}{2}=\Theta(N^2)$.

QUESTION 2: [20 points]

a) What is the form of the input array that triggers the best case of the insertion sort? [5 points]

Best case scenario of the insertion sort happens when the input array is already sorted

b) What is the complexity of this best–case behavior in Θ notation? [5 points]

The complexity of this best case behavior is Θ(N)

c) Explain how this particular form of the array results in this complexity. [10 points]

When the input array is already sorted, there will be only 1 comparison between the current element and the next element to verify whether the current and next elements are ordered by themselves. Thus, if the input array is already sorted, it is guaranteed that there will be only 1 comparison for each iteration in the array. If we have an array of N elements, there will be a total of $N-1$ comparisons (there won't be a comparison for the last element).

- $N-1 = O(N)$ because $N-1 \leq cN$ is true when $c = 1$ and $N \geq n_0$ for $n_0 \geq 1$.
- $N-1 = \Omega(N)$ because $N-1 \geq cN$ is true when $c = \frac{1}{2}$ and $N \geq n_0$ for $n_0 \geq 1$.
- Since $N-1 = O(N)$ and $N-1 = \Omega(N)$, it means that $N-1 = \Theta(N)$.

QUESTION 3: [50 points]

Suppose that you are trying to prove $(5n+4)^2 = O(n^2)$ by using the formal definition of O-notation, where $n \geq 0$.

In order to show that $(5n+4)^2 = O(n^2)$ by using the formal definition of O-notation, we need to pick constants c and n_0 such that for any $n \geq n_0$ we have

$$(5n+4)^2 \leq cn^2$$

a) If you use $n_0 = 2$, what is the smallest c value that makes the proof go through? [25 points]

If we simplify this equation we are having $25n^2 + 40n + 16 \leq cn^2 = (25-c)n^2 + 40n + 16 \leq 0$. Since the quofficiant of the highest order number(n^2) is $(25-c)$. After a while, the equation will always be satisfied if c $\geq$ 26.

But if $n_0 = 2$ it means that n can take any value greater than 2. Since n^2 will dominate at big numbers we can find the value of c if we assign the smallest possible number(2) to n.

- the equation is $(5n+4)^2 \leq cn^2$ if we assign 2 to n we get $(5*2+4)^2 \leq c*4$ which gives us $c \geq 49$. So the smallest value of c is 49.

b) If you use c = 36, what is the smallest n_0 value that makes the proof go through? [25 points]

- if we put 36 to the value of c the equation coming from formal definition of O-notation becomes $(5n+4)^2 \leq 36n^2$.
- if we simplify this equation it becomes $25n^2 + 40n + 16 \leq 36n^2 = -11n^2 + 40n + 16 \leq 0 = (4-n)(11n+4) \leq 0$
- so the critical points are 4 and $-\frac{4}{11}$. According to this we can conclude that equation becomes negative when n is in (-∞, $-\frac{4}{11}$], it is positive when n is in ($-\frac{4}{11}$,4) and it is negative when n is in $[4, \infty)$. We need to choose a negative point because of the final simplified equation and n_0 should be greater than or equal to some number not smaller or equal, that's why n_0 should be 4.

QUESTION 4: [10 points]

a) Rank the following functions in descending order with respect to their growth rates. [10 points]

$$lg(n!) \quad n! \quad n2^n \quad (2^2)^n \quad lg^2 n \quad (lgn)!$$

When those functions are ordered in descending order the arrangement becomes;

$$2^{2n} > n! > n2^n > lg(n!) > lg(n!) > lg^2 n$$