Assignment 2

Somesh Devagekar,Protyush Kumar Das

Hochschule Bonn-Rhein-Sieg

somesh.devagekar@smail.inf.h-brs.de
protyush.das@smail.inf.h-brs.de

October 22, 2018

1.¿Why is a special notation needed to classify algorithms?

Algorithms put the science in computer science,and finding good algorithms and knowing when to apply them will allow you to write interesting and important programs.The more complex algorithms, more the effort to make the program faster. How do we measure efficiency? We could time how long it takes to run the code, but that would tell us on a paticular programming language, a paticular processor with a paticular input given.So instead of this a generalised form was developed called Asymptotic analysis. The runtime for algorithm depends on how long it takes a computer to run the code, also it shows how fast a function grows with input size or rate of growth of a running time.

Explanation

Lets say an algorithm runs on a input size $6n^2 + 1000n + 130$. therefore the running time grows as n^2. Here if we neglect the coeficients, then 100n+130. therefore by dropping he less significant terms and the constant coefficients, we can focus on the important part of an algorithm's running time i.e. asymptotic notation and types are big- notation, big-O notation, and big- notation. Big O specifically describes the worst-case scenario, and can be used to describe the execution time.

Reference:

- khanacademy.com

question 2.1

Prove that T(n)=$100n^2$ is

ANS By proof T(n)is O($g(n)$)if T(n)$\leq c.g(n)$ for some $n \geq n_0$

So T(n)=O(n^2) if T(n)$\leq c.n^2, \forall n \geq n_0$ given c,n$\geq$0

Hence assuming $100n^2 \leq c.n^2$

or, $100n^2 \div n^2 \leq c.n^2 \div n^2$ (dividing both sides by n^2)

or, 100$\leq c$ which holds that the equation is valid $\forall$ c$\geq$100

Therefore it proved that T(n)=$100n^2$ $\forall$ c$\geq$100

The values of n doesnot affect the coefficient of c.

question 2.2

Prove that T(n)=$n^6 + 100n^5$ O(n^6)

ANS Following the Big O definition as the above solved euation it can be said that

$n^6 + 100n^5 \leq c.n^6$

or, $1 + \frac{100}{n} \leq c$ (dividing both sides by n^6)

Here Big-O holds for $n \geq n_0 = 1$ and $c \leq (1 + 100)$

larger the value of n results in smaller factors of c, thus the above statement is true.

question3

```
1.sum = 0
2.for i in range(0, J):
3.      for j in range(0, K):
4.            if arr[i][j] ≤ ANYCONST:
5.            sum = sum + arr[i][j]
6. print(sum)
```

ANS Time Complexity:

1=O(1)

2=O(n)

3=O(n)

4=O(1)

5=O(1)

6=O(1)

Result=O(1)+O(n)*O(n)*(O(1)+O(1))+O(1)

or, Result=O(1)+O($2n^2$)+O(1) since, O($n * n * (1 + 1)$)=O($2n^2$)

or, Result=O($2n^2$) or, for worst case Result=O(n^2)

Conversion:-

1sec=10^6µs

1min=6x10^7µs

1hour=3.6x10^9µs

1day=8.64x10^{10}µs

1month=2.592x10^{12}µs

1year=3.1557x10^{13}µs

1centuary=3.1557x10^{15}µs

Calculations:-

n!	n!=10^6=9
log(n)	10^6=n=$2^{10^\wedge(6)}$
nlog(n)	10^6=n=62746
n	10^6
n^2	n^2=10^3
n^3	n^3=10^2
2^n	2^n=10^6=6xlog10=19
sqrt(n)	10^6=n=10^{12}

Result:-

f(n)	1sec	1min	1hour	1day	1month	1year	1centuary
log(n)	≈ 10^{300000}	≈ $10^{1080000000}$	2.36x10	$2^{8.64x10^\wedge 8}$	≈ $10^{7.7x10(11)}$	$2^{3.19x10^\wedge(9)}$	≈$10^{9.332x10^\wedge(14)}$
sqrt(n)	10^{12}	3.6x10^{15}	1.2x10^{19}	7.46x10^{21}	6.71x10^{24}	9.94x10^{26}	9.945x10^{30}
n	10^6	6x10^7	3.6x10^9	8.64x10^{10}	2.592x10^{12s}	3.1557x10^{13}	3.1557x10^{15}
n^2	1000	7745	60000	293938	1609968	5615692	56175382
n^3	100	391	1532	4420	13736	31593	146677
2^n	19	25	31	36	41	44	51
n!	9	11	12	13	15	16	17
nlogn	62746	2801418	133378059	2755147513	71870656400	7.9x10^{11}	9.9x10^{30}