Assignment 2

Somesh Devagekar,Protyush Das

Hochschule Bonn-Rhein-Sieg

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

October 18, 2018

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 you 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 computer with a paticular input given.So instead of this a generalised form was developed called Asymptotic analisis. 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.

question 2.1

T(n)=$100n^2$ O(n^2)
By proof T(n)is O(n^2)if T(n)$\leq c.n^2$ for some $n \geq n_0$
for values of $n \geq n_0 = 1$
therefore it can hold as there is no constant factor, Larger values of n0 result in smaller factors n

question 2.2

T(n)=$n^6 + 100n^5$ O(n^6)
therefore $\frac{1}{n} + \frac{100}{n^5} \leq c$
Here Big-O holds for $n \geq n_0 = 1$ and $c \leq 101(1 + 100)$
larger the value of n results in smaller factors of n, thus the above statement is true.

question3

sum $= 0$
for i in range(0, J):
for j in range(0, K):
$if arr[i][j] \leq ANYCONST$:
sum $=$ sum $+$ arr[i][j]
print(sum)
Result:On^2