Adish Betawar

CS 146

Section 4

Group 1

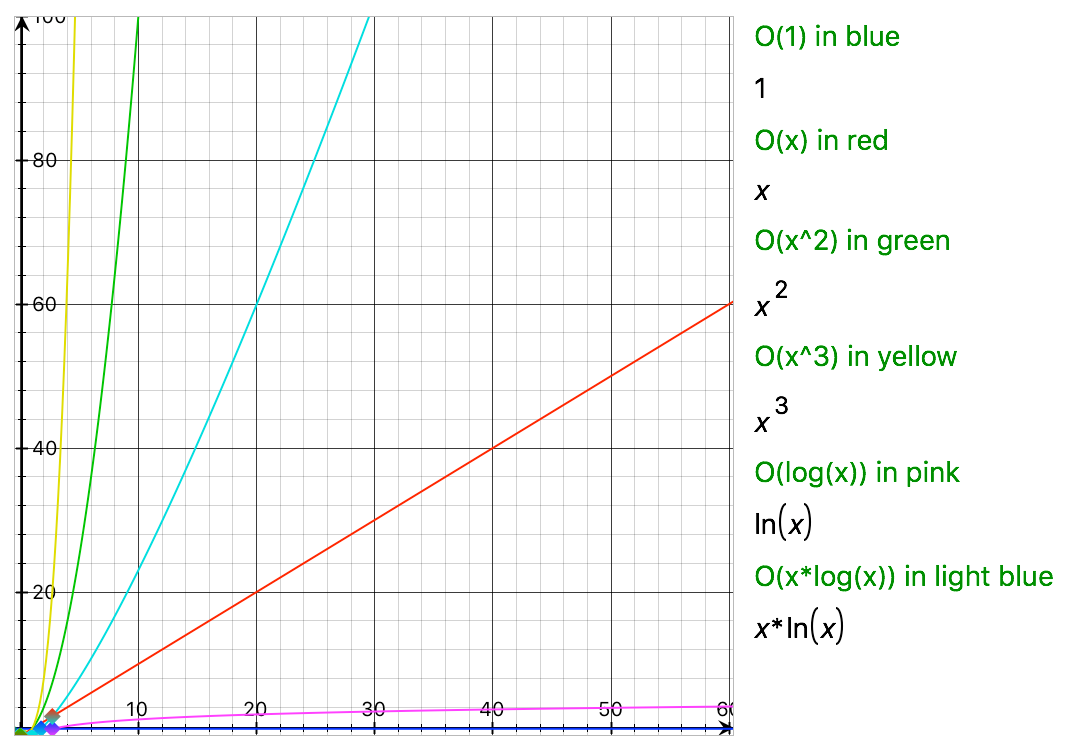
2/28/2016

Homework #1

1. Big-O Notation

O(1) O(n) O(n^2) O(n^3) O(logn ) O(nlogn)

N = 100, 1000, 10000, 100000, 1000000



* 1. Graph N = 0 to 100, shows that the O(N) function rate of growth

T = Runtime in seconds (s)

N = Number of Entries

Figure 1.1

Table 1.1

|  | O(1) | O(n) | O(n^2) | O(n^3) | O(log(n)) | O(n\*log(n)) |
| --- | --- | --- | --- | --- | --- | --- |
| 100 | 1 | 10^2 | 10^4 | 10^6 | 6.6 | 6.6\*10^2 |
| 1000 | 1 | 10^3 | 10^6 | 10^9 | 10 | 10\*10^3 |
| 10,000 | 1 | 10^4 | 10^8 | 10^12 | 13 | 13\*10^4 |
| 100,000 | 1 | 10^5 | 10^10 | 10^15 | 17 | 17\*10^5 |
| 1,000,000 | 1 | 10^6 | 10^12 | 10^18 | 20 | 20\*10^6 |

* 1. Binary search Vs. Linear Search
     + Binary search
       - Starts a search at the middle of a **sorted list**
       - Checks wether the middle value is less then or greater than the value you are searching for

| Array | ArrayList |
| --- | --- |
| Stores primitive class objects and data types | It can only store objects |
| In java, it is defined as a fundamental data structure | It is a collection framework |
| Stores similar types of data | Ability to store data of many types |
| Fixed array size | Can be dynamically reallocated |
| A class can contain an array | A class consisting of many methods |
| Operations are completed in constant time | Only a few operations are performed in constant time |

* + - * + If less then desired value

Searches to the left

Otherwise searches to the right

* + - * Process repeats itself until we find the value
      * Big-O efficiency is O(log(N))
    - Linear Search
      * Looks down a list one element at a time
      * Big-O efficiency is O(N)

C. Arrays vs ArrayList

Arrays:

get(): O(1)

deletion(): O(1)

Find: O(n) -> n.length of the array

ArrayList:

Get(): O(1)

remove(): O(1)

Find(): O(1)

D. Analyze three sorting methods

- Insertion sort

- The time complexity is O(n^2)

-Selection sort

- The time complexity is O(n^2)

-Quick sort

-The time complexity depends on the circumstance in which the input list is sorted and the left element is the pivot point, the Worst case time complexity is O(n^2). On the other hand, if the pivot is randomly chosen the time complexity would be O(n log n).

E. Generics and examples

4. Code for Tower of Hanoi on attached file.

- Big-O notation for Tower of Hanoi is = O(2^n)

Calculations:

T(n) = T(n-1) + c + T(n-1) = 2T(n-1) + c = 2(2(T(n-2)+c)+c) = 2^(n-1) T(1) + c2^(n-2) + … + c2+c = c(2^n)-1 = O(2^n)

- Greedy algorithm analysis is a process that determines simple solutions to difficult problems with many steps by calculating which step will generate the best result. The greedy algorithm works recursively in constructing a set of objects from smallest possible parts.The algorithm focuses more on the outputs for smaller problems than the problem as a whole. The benefit of using this method is the problem can become much easier to comprehend.