

Java-Success

Java developer training with a difference - [650+ Java interview questions & answers](#), Java tools to get the job done, and How in Java? with lots of code, diagrams, tutorials, top tips, & [16 key areas](#) for your Java interview, on the job, & career success.

Search this blog:

or Google with **Java-Success** in search.
e.g "Java-success Spring"

[200+ Java Interview Questions Answered](#)

[Core Java Q&A](#)

[Coding Q&A](#)

[JEE Q&A](#)

[Java Architect Q&A](#)

[Open Q&A](#)

[Other Q&A](#)

[Prepare?](#)

[Testing Q&A](#)

[Core Java Tutorial](#)

[Java EE Tutorial](#)

[Top X list](#)

[Java Articles](#)

[Career Tips](#)

[Java Freelancer](#)

[My Books](#)

[Java-Success?](#)

Apr 14, 2014

Understanding Big O notations through Java examples

Q. Have you seen job advertisements requiring Java candidates to work in **real-time** or high volume transaction processing systems?

If you are applying for such jobs, you can be quizzed on **Big O** notation. Here are some basics to brush up on.

Big-O gives you the upper bound. For example, if you need to search an element in an array and you expect the array to be large, you might just say that you opt for a binary search instead of a sequential scan because the former has $O(\log n)$ complexity whereas the latter has $O(n)$ complexity.

Big-O	Description/Example	If n=10, and c=2	If n=20, and c=2
O(1) Constant	Running time is constant. Determining if a String is equal to a given value <pre> 1 if(str.equals("java")) 2 { 3 return true; 4 } 5 else { 6 return false; 7 } </pre> A Map look up by key -- map.get(key);	1	1
O(log n) Logarithmic	Running time increases logarithmically in proportion to the input size. Finding an item in a sorted array with a binary search For example, search for 2 in a list of numbers 1,2,3,4,5,6,7 <ul style="list-style-type: none"> Step 1: Sort the data set in ascending order as binary search works on sorted data. Step 2: Get the middle element (e.g. 4) of the data set and compare it against the search item (e.g. 2), and if it is equal return that element Step 3: If search item value is lower, discard the second half of the data set and use the first half (e.g. 1,2,3). If the search item value is higher, then discard the first half and use the second half (e.g. 5,6,7) Step 4: Repeat steps 2 and 3 until the search item is found or the last element is reached and search item is not found. So, it is iteratively reducing the number of elements it process.	$\log 10 = 1$	$\log 20 = 1.301$

Stay connected to Java-Success

Arulkumaran K...

g+ Follow

1,082 followers

241

Page views

g+1

4,011,9

5 recent Java developer posts

- [Java with SVNKit to programmatic checkout and update files](#)
- [Java and XML tutorial with DOM P:](#)
- [11 tips to writing low latency real-ti applications in Java](#)
- [6 tips to writing loosely coupled Jav applications with examples](#)
- [Which Jar has a particular class? or from which jar a class was loaded? solving jar hell issues](#)

O(n) Linear	<p>Running time increases in direct proportion to the input size</p> <p>Finding an item in an unsorted array or list.</p> <pre> 1 for(int i = 0; i < strings.Length; i++) { 2 if(strings[i].equals("java")) 3 { 4 return true; 5 } 6 else { 7 return false; 8 } 9 }</pre>	10	20
O(n log n) Super linear	<p>Running time is midway between a linear algorithm and a polynomial algorithm</p> <p>Collections.sort is an optimized merge sort which actually guarantees O(n log n). A quicksort is generally considered to be faster than a merge sort (i.e. n log n) but isn't stable and doesn't guarantee n log(n) performance. For Merge sort worst case is O(n*log(n)), for Quick sort: O(n^2).</p>	10 log 10 = 10	20 log 20 = 26.02
O(n^c) Polynomial	<p>Running time grows quickly based on the size of the input.</p> <p>O(n^2) Quadratic -- bubble Sort (worst case or naive implementation)</p> <pre> 1 for(int i = 0; i < strings.Length; i++){ 2 for(int j = 0; j < strings.Length; j++){ 3 if(i == j) // Don't compare with self 4 { 5 continue; 6 } 7 8 if(strings[i].equals(strings[j])) 9 { 10 return true; 11 } 12 else { 13 return false; 14 } 15 } </pre>	10^2 = 100	20^2 = 400
O(c^n) Exponential	<p>Running time grows even faster than a polynomial algorithm.</p> <p>Recursive computation of Fibonacci numbers is a good example of O(2^n) algorithm</p> <pre> 1 public int fib(int n) { 2 if (n <= 1) return n; 3 else return fib(n - 2) + fib(n - 1); 4 }</pre>	2^10 = 1,024	2^20 = 1,048,576
O(n!) Factorial	<p>Running time grows the fastest and becomes quickly unusable for even small values of n.</p> <p>Recursive computation of factorial</p> <pre> 1 public void nFactorial(int n) { 2 for(int i=0; i<n; i=n-1) { 3 nfactorial(i); 4 } </pre>	10! = 10*9*8*7*6*5*4*3*2*1= 3,628,800	20!= 2432902008176640000

Top 10 Java developer training p

[Java Web Services Interview Quest and Answers: Overview](#)

[30+ Core Java multithreading inter questions and answers](#)

[15+ Hibernate Interview Questions Answers: Overview](#)

[Top 20+ Java EE interview questio and answers that experienced Java developers must know](#)

[Core Java coding questions frequer asked in written tests and interview part 1: Immutability and object references](#)

[Java Collection Interview Question Answers](#)

[Selenium and Web Driver Interview Questions and Answers](#)

[20+ Spring Interview Questions an Answers: Overview](#)

[Article: Sample Java Resume or CV](#)

[Java OO Interview Questions and Answers](#)

Java companion and essentials F

[Free Sample Core Java & EE Interview Questions & Answers PDF downloads](#)

Programmer Job hunting & marketing

Programming Career Success
5 reasons why I still program, and not get into architecture or managerial roles?



+2 Recommend this on Google

Labels: [Coding](#), [Performance](#)

No comments:

Post a Comment