Register | Login | Logout | Contact Us

search here … Go

Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › Tech Key Areas › Written Tests › Java Data Structures & Algorithms › Big O notation › 04: Understanding Big O notations through Java examples

04: Understanding Big O notations through Java examples

Posted on November 22, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

0 Like Share 0 G+1 0 Share

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 wheres the latter has O(n) complexity.

<table>
<tr><th>Big-O</th><th>Description/Example</th></tr>
<tr><td>O(1)

Constant</td><td><b>Running time is constant.</b>

Determining if a String is equal to a given value<pre>
if(str.equals("java"))
{
    return true;
}
else {
    return false;
</pre></td></tr>
</table>

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all
- Ice Breaker Interview Q&A (
- Core Java Interview Q&A (1
  - Java Overview (4)
  - Data types (6)
  - constructors-methods-init
  - Reserved Key Words (2)
  - Classes (3)
  - Objects (8)
  - OOP (10)
  - GC (2)
  - Generics (5)
  - FP (8)
  - IO (7)
  - Multithreading (12)
  - Algorithms (5)
  - Annotations (2)
  - Collection and Data struc
    - ♦ Find the first non rep
    - ♦ Java Collection over
    - ♥ Java Iterable Vs Itera
    - ♥♦ HashMap & HashS
    - ♦ Sorting objects in Ja
    - 02: ♦ Java 8 Streams,
    - 04: Understanding Big
    - 4 Java Collection Quiz
    - If Java did not have a s
    - Java 8: Different ways
    - Part-3: Java Tree struc
    - Sorting a Map by value
    - When to use which Ja
  - Differences Between X &

```
}

```

A Map look up by key — map.get(key);

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

- 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.

O(n)

Linear

Running time increases in direct proportion to the input size

Finding an item in an unsorted array or list.

```
for(int i = 0; i < strings.Length; i++) {
  if(strings[i].equals("java"))
    {
   return true;
    }
    else {
   return false;
    }
}

```

⊞ Event Driven Programmi
⊞ Exceptions (2)
⊞ Java 7 (2)
⊞ Java 8 (24)
⊞ JVM (6)
⊞ Reactive Programming (
⊞ Swing & AWT (2)
⊞ JEE Interview Q&A (37)
⊞ Pressed for time? Java/JEE
⊞ SQL, XML, UML, JSON, Re
⊞ Hadoop & BigData Intervie
⊞ Java Architecture Interview
⊞ Scala Interview Q&As (13)
⊞ Spring, Hibernate, & Maven
⊞ Testing & Profiling/Samplin
⊞ Other Interview Q&A for Ja
⊞ ▶Free Java Interview Vide

16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3)
⊞ Java Debugging (21)
⊞ Judging Experience Intervie
⊞ Low Latency (7)
⊞ Memory Management (7)
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Management (

80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2)
⊞ Akka Tutorial (9)

O(n log n) Super linear	**Running time is midway between a linear algorithm and a polynomial algorithm** 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).		
O(n^c) Polynomial	**Running time grows quickly based on the size of the input.** O(n^2) Quadratic — bubble Sort (worst case or naive implementation)		

```
for(int i = 0; i < strings.Length; i++){
  for(int j = 0; j < strings.Length; j++)
      if(i == j) // Don't compare with self
   {
  continue;
   }

     if(strings[i].equals(strings[j]))
     {
     return true;
     }
     else {
    return false;
    }
}


```

O(c^n) Exponential	**Running time grows even faster than a polynomial algorithm.** Recursive computation of Fibonacci numbers is a good example of O(2^n) algorithm	1,024	

```
public int fib(int n) {
    if (n <= 1) return n;
    else return fib(n - 2) + fib(n - 1);
}

```

O(n!) Factorial	**Running time grows the fastest and becomes quickly unusable for even small values of n.** Recursive computation of factorial	10! = 10*9*8*7*6*5*4*3*2*1= 3,628,800	20!= 2432902008176640000

⊞ Core Java Tutorials (20)
⊞ Hadoop & Spark Tutorials (2
⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tutorial
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
⊞ Can you write code? (22)
⊞ ♦ Complete the given code
⊞ Converting from A to B (6)
⊞ Designing your classes & in
⊞ Java Data Structures & Algo
⊞ Passing the unit tests (5)
⊞ What is wrong with this cod
⊞ Writing Code Home Assigne
⊞ Written Test Core Java (3)
⊞ Written Test JEE (1)

How good are your .....?

open all | close all
⊞ Career Making Know-hows
⊞ Job Hunting & Resume Writ

```java
public void nFactorial(int n) {
  for(int i=0; i<n; i=n-1) {
    nfactorial(i);
  }

```

Popular Posts

- ♦ 11 Spring boot interview questions & answers

  823 views
- ♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

  765 views
- 18 Java scenarios based interview Questions and Answers

  399 views
- 001A: ♦ 7+ Java integration styles & patterns interview questions & answers

  388 views
- 01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

  295 views
- ♦ 7 Java debugging interview questions & answers

  293 views
- 01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

  285 views
- ♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

  279 views
- ♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

  239 views
- 001B: ♦ Java architecture & design concepts interview questions & answers

  201 views

Bio | Latest Posts

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ 07: 5 Basic multi-threading interview questions & answers

Java coding question on recursion and generics ›

Posted in Big O notation, Collection and Data structures, Low Latency, member-paid

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function? ☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑