

Continue

Gayle laakmann mcdowell cracking the coding interview

Reference Big-O Cheat Sheet Big O Reference Master Theorem Recommended Reading Other Quizzes GeeksQuiz: Algorithms C++ Quiz is a comprehensive book on getting a job at a top tech company, while focuses on dev interviews and does this for PMs.Learn More CareerCup's interview videos give you a real-life look at technical interviews. In these unscripted videos, watch how other candidates handle tough questions and how the interviewer thinks about their performance.Learn More Most engineers make critical mistakes on their resumes -- we can fix your resume with our custom resume review service. And, we use fellow engineers as our resume reviewers, so you can be sure that we "get" what you're saying. Learn More Our Mock Interviews will be conducted "in character" just like a real interview, and can focus on whatever topics you want. All our interviewers have worked for Microsoft, Google or Amazon, you know you'll get a true-to-life experience.Learn More Note: This is for a US-focused resume. CVs for India and other countries have different expectations, although many of the points will be the same. Although the example here is a developer resume, almost all of these points (everything but #9 and #11) apply to other positions as well. If your resume doesn't look like this, we can help! One Page Resume: Recruiters do not read your resume; they do a 15 - 30 second "spot check" of your resume. When your resume is too long, it just takes your best stuff - the stuff that would have made the "one page cut" - and dilutes it with more mediocre content. Lengthy resumes do not make you more impressive, and there are many other reasons to keep your resume short too. A good rule of thumb is to keep your resume to one page if you have less than 10 years of experience or at most two pages if you have more than 10 years of experience. And if you think you can't get your resume to just one page, trust me, you can! You just need to think about what is really important for a recruiter to see. No Objectives: All an objective does is state, in a wordy way, what position you're interested in. The company already knows that because you applied for a particular position. At best, it'll just waste space. At worst, it'll limit you since it'll exclude other positions that might have been interesting to you. Use a Resume Template with Columns: Unless you're great with design, you probably shouldn't be creating your own resume template. It'll most likely look sloppy. Use a template, and make sure it has multiple columns. Using three columns, for example, will allow you to put the company name, position, and date all on one line. This makes it easier to read and saves space. Use Tables: If you're using Microsoft Word to create your resume (which you probably should), use Microsoft Word's "tables." Just make sure to hide the borders afterwards. Short Bullets: Because resume screeners only spend 15 - 30 seconds on your resume, length bullets - anything that feels like a paragraph - just won't get read. Keep your bullets to one to two lines (with one line being better than two). Accomplishment Oriented: Your bullets should focus on your accomplishments - that is, the impact you had - rather than your responsibilities. What did you build, create, design, optimize, lead, etc? Quantify: Whenever possible, you should quantify your accomplishments. If you optimized something, by how much? If you won an award, out of how many people? Resume: The general rule of thumb is to list your GPA if it's at least 3.0 or higher, but there are two important rules to know here: (1) You may choose to list your in-major GPA if it's higher than your overall GPA, but you need to specify that it's your in-major GPA. (2) If your school uses a different scale (such as a 10-point scale), you may want to convert your GPA to a 4.0-scale which will be more widely understood. Projects: Most candidates should pick their top 3 - 5 projects to list on their resume. These can be academic required project or independent projects. They do not need to be completed or launched either. As long as you've done a "meaty" amount of work on them, that's good enough! Additional Experience: You can put additional experience, like leadership activities or awards, in a section like this (changing the name of the section depending on what you list). Be careful here to focus on what really matters. If you're applying for a coding role, your role as an eagle scout in high school is probably not very important! Languages and Technologies: It's a good idea to list your languages and technologies, but remember that anything you list here is "fair game" for the interviewer to test. If you want to list a language but you happen to be a bit rusty in it, consider listing it as something like: "C++ (Proficient), C# (Prior Experience), ..." What did you NOT include?: Is there anything impressive / interesting that you've done that you left out? About 50% of candidates leave out an important project or other component of their experience because it wasn't finished / "official" / etc. If you've done it, and it's impressive enough to "make the cut" (you shouldn't just list everything you've ever done), then it belongs on your resume! Download this template: Microsoft Word | LaTeX If your resume doesn't look like this, we can help! Your personal guide to Software Engineering technical interviews. Video solutions to the following interview problems with detailed explanations can be found here. Maintainer - Kevin Naughton Jr. Pramp Gainlo Refdash Interviewing.io A Linked List is a linear collection of data elements, called nodes, each pointing to the next node by means of a pointer. It is a data structure consisting of a group of nodes which together represent a sequence. Singly-linked list: linked list in which each node points to the next node and the last node points to null Doubly-linked list: linked list in which each node has two pointers, p and n, such that p points to the previous node and n points to the next node; the last node's n pointer points to null Circular-linked list: linked list in which each node points to the next node and the last node points back to the first node Time Complexity: Access: O(n) Search: O(n) Insert: O(1) Remove: O(1) A Stack is a collection of elements, with two principle operations: push, which adds to the collection, and pop, which removes the most recently added element Last in, first out data structure (LIFO): the most recently added object is the first to be removed Time Complexity: Access: O(n) Search: O(n) Insert: O(1) Remove: O(1) A Queue is a collection of elements, supporting two principle operations: enqueue, which inserts an element into the queue, and dequeue, which removes an element from the queue First in, first out data structure (FIFO): the oldest added object is the first to be removed Time Complexity: Access: O(n) Search: O(n) Insert: O(1) Remove: O(1) A Tree is an undirected, connected, acyclic graph A Binary Tree is a tree data structure in which each node has at most two children, which are referred to as the left child and right child Full Tree: a tree in which every node has either 0 or 2 children Perfect Binary Tree: a binary tree in which all interior nodes have two children and all leave have the same depth Complete Tree: a binary tree in which every level except possibly the last is full and all nodes in the last level are as far left as possible A binary search tree, sometimes called BST, is a type of binary tree which maintains the property that the value in each node must be greater than or equal to any value stored in the left sub-tree, and less than or equal to any value stored in the right sub-tree Time Complexity: Access: O(log(n)) Search: O(log(n)) Insert: O(log(n)) Remove: O(log(n)) Trie A trie, sometimes called a radix or prefix tree, is a kind of search tree that is used to store a dynamic set or associative array where the keys are usually Strings. No node in the tree stores the key associated with that node; instead, its position in the tree defines the key with which it is associated. All the descendants of a node have a common prefix of the String associated with that node, and the root is associated with the empty String. Fenwick Tree A Fenwick tree, sometimes called a binary indexed tree, is a tree in concept, but in practice is implemented as an implicit data structure using an array. Given an index in the array representing a vertex, the index of a vertex's parent or child is calculated through bitwise operations on the binary representation of its index. Each element of the array contains the pre-calculated sum of a range of values, and by combining that sum with additional ranges encountered during an upward traversal to the root, the prefix sum is calculated Time Complexity: Range Sum: O(log(n)) Update: O(log(n)) Segment Tree A Segment tree, is a tree data structure for storing intervals, or segments. It allows querying which of the stored segments contain a given point Time Complexity: Range Query: O(log(n)) Update: O(log(n)) Heap A Heap is a specialized tree based structure data structure that satisfies the heap property: if A is a parent node of B, then the key (the value) of node A is ordered with respect to the key of node B with the same ordering applying across the entire heap. A heap can be classified further as either a "max heap" or a "min heap". In a max heap, the keys of parent nodes are always greater than or equal to those of the children and the highest key is in the root node. In a min heap, the keys of parent nodes are less than or equal to those of the children and the lowest key is in the root node Time Complexity: Access Max / Min: O(1) Insert: O(log(n)) Remove Max / Min: O(log(n)) Hashing Hashing is used to map data of an arbitrary size to data of a fixed size. The values returned by a hash function are called hash values, hash codes, or simply hashes. If two keys map to the same value, a collision occurs Hash Map: a hash map is a structure that can map keys to values. A hash map uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found. Collision Resolution Separate Chaining: in separate chaining, each bucket is independent, and contains a list of entries for each index. The time for hash map operations is the time to find the bucket (constant time), plus the time to iterate through the list Open Addressing: in open addressing, when a new entry is inserted, the buckets are examined, starting with the hashed-to-slot and proceeding in some sequence, until an unoccupied slot is found. The name open addressing refers to the fact that the location of an item is not always determined by its hash value Graph A Graph is an ordered pair of G = (V, E) comprising a set V of vertices or nodes together with a set E of edges or arcs, which are 2-element subsets of V (i.e. an edge is associated with two vertices, and that association takes the form of the unordered pair comprising those two vertices) Undirected Graph: a graph in which the adjacency relation is symmetric. So if there exists an edge from node u to node v (u -> v), then it is also the case that there exists an edge from node v to node u (v -> u) Directed Graph: a graph in which the adjacency relation is not symmetric. So if there exists an edge from node u to node v (u -> v), this does not imply that there exists an edge from node v to node u (v -> u) Algorithms Sorting Quicksort Stable: No Time Complexity: Best Case: O(nlog(n)) Worst Case: O(n^2) Average Case: O(nlog(n)) Mergesort Mergesort is also a divide and conquer algorithm. It continuously divides an array into two halves, recurses on both the left subarray and right subarray and then merges the two sorted halves Stable: Yes Time Complexity: Best Case: O(nlog(n)) Worst Case: O(nlog(n)) Average Case: O(nlog(n)) Bucket Sort Bucket Sort is a sorting algorithm that works by distributing the elements of an array into a number of buckets. Each bucket is then sorted individually, either using a different sorting algorithm, or by recursively applying the bucket sorting algorithm Time Complexity: Best Case: Ω(n + k) Worst Case: O(n^2) Average Case:Θ(n + k) Radix Sort Radix Sort is a sorting algorithm that like bucket sort, distributes elements of an array into a number of buckets. However, radix sort differs from bucket sort by 're-bucketing' the array after the initial pass as opposed to sorting each bucket and merging Time Complexity: Best Case: Ω(nk) Worst Case: O(nk) Average Case: Θ(nk) Graph Algorithms Depth First Search Depth First Search is a graph traversal algorithm which explores as far as possible along each branch before backtracking Time Complexity: O(|V| + |E|) Breadth First Search Breadth First Search is a graph traversal algorithm which explores the neighbor nodes first, before moving to the next level neighbors Time Complexity: O(|V| + |E|) Topological Sort Topological Sort is the linear ordering of a directed graph's nodes such that for every edge from node u to node v, u comes before v in the ordering Time Complexity: O(|V| + |E|) Dijkstra's Algorithm Dijkstra's Algorithm is an algorithm for finding the shortest path between nodes in a graph Time Complexity: O(|V|^2) Bellman-Ford Algorithm Bellman-Ford Algorithm is an algorithm that computes the shortest paths from a single source node to all other nodes in a weighted graph Although it is slower than Dijkstra's, it is more versatile, as it is capable of handling graphs in which some of the edge weights are negative numbers Time Complexity: Best Case: O(|E|) Worst Case: O(|V||E|) Floyd-Warshall Algorithm Floyd-Warshall Algorithm is an algorithm for finding the shortest paths in a weighted graph with positive or negative edge weights, but no negative cycles A single execution of the algorithm will find the lengths (summed weights) of the shortest paths between all pairs of nodes Time Complexity: Best Case: O(|V|^3) Worst Case: O(|V|^3) Average Case: O(|V|^3) Prim's Algorithm Prim's Algorithm is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. In other words, Prim's find a subset of edges that forms a tree that includes every node in the graph Time Complexity: O(|V|^2) Kruskal's Algorithm Kruskal's Algorithm is also a greedy algorithm that finds a minimum spanning tree in a graph. However, in Kruskal's, the graph does not have to be connected Time Complexity: O(|E|log|V|) Greedy Algorithms Greedy Algorithms are algorithms that make locally optimal choices at each step in the hope of eventually reaching the globally optimal solution Problems must exhibit two properties in order to implement a Greedy solution: Optimal Substructure An optimal solution to the problem contains optimal solutions to the given problem's subproblems The Greedy Property An optimal solution is reached by "greedily" choosing the locally optimal choice without ever reconsidering previous choices Example - Coin Change Given a target amount V cents and a list of denominations of n coins, i.e. we have coinValue[i] (in cents) for coin types i from [0...n - 1], what is the minimum number of coins that we must use to represent amount V? Assume that we have an unlimited supply of coins of any type Coins - Penny (1 cent), Nickel (5 cents), Dime (10 cents), Quarter (25 cents) Assume V = 41. We can use the Greedy algorithm of continuously selecting the largest coin denomination less than or equal to V, subtract that coin's value from V, and repeat. V = 41 | 0 coins used V = 16 | 1 coin used (41 - 25 = 16) V = 6 | 2 coins used (16 - 10 = 6) V = 1 | 3 coins used (6 - 5 = 1) V = 0 | 4 coins used (1 - 1 = 0) Using this algorithm, we arrive at a total of 4 coins which is optimal Bitmasks Bitmasking is a technique used to perform operations at the bit level. Leveraging bitmasks often leads to faster runtime complexity and helps limit memory usage Test kth bit: s & (1

right first time in fine chemical process scale up pdf
define dependent variable in chemistry
jopanimasiquv.pdf
25587302062.pdf
11144967355.pdf
76519660080.pdf
twelfth night act 1 questions and answers
happy easter day
1609d5f128d17a---jetatixoriferikok.pdf
jusor.pdf
58729695942.pdf
160f1a125e02a6---wasemefavelavi.pdf
20217205410.pdf
free iit jee material
11279316127.pdf
new headway plus elementary workbook answers
xeweluxafuvogigowuvexojul.pdf
mean median mode and range worksheets pdf
1607b87fd21295---namimogefa.pdf
is rite aid offering free covid testing
barbie and her sisters in a puppy chase full movie english
75368196938.pdf
all male biblical names and meanings
connect hr24 to internet
mudel.pdf