**System Design**

Main:

<https://github.com/checkcheckzz/system-design-interview>

Basics:

<http://www.aosabook.org/en/distsys.html>

Graph:

<https://www.facebook.com/notes/facebook-engineering/under-the-hood-indexing-and-ranking-in-graph-search/10151361720763920>

https://code.facebook.com/posts/1737605303120405/dragon-a-distributed-graph-query-engine/

https://www.facebook.com/notes/facebook-engineering/tao-the-power-of-the-graph/10151525983993920/

https://www.youtube.com/watch?v=QHkhyY9atkE

https://engineering.linkedin.com/real-time-distributed-graph/using-set-cover-algorithm-optimize-query-latency-large-scale-distributed

Scale:

<https://blog.twitter.com/2017/the-infrastructure-behind-twitter-scale>

Network

https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

https://blog.twitter.com/2017/the-infrastructure-behind-twitter-scale

Backup Servers:

https://code.facebook.com/posts/1007323976059780/continuous-mysql-backup-validation-restoring-backups/

News Feed:

<https://www.slideshare.net/danmckinley/etsy-activity-feeds-architecture>

http://highscalability.com/blog/2013/7/8/the-architecture-twitter-uses-to-deal-with-150m-active-users.html

https://engineering.linkedin.com/blog/2016/03/followfeed--linkedin-s-feed-made-faster-and-smarter

<http://blog.gainlo.co/index.php/2016/03/29/design-news-feed-system-part-1-system-design-interview-questions/>

Search Engines:

<http://www.ardendertat.com/2012/01/11/implementing-search-engines/>

<http://softwareengineering.stackexchange.com/questions/38324/how-would-you-implement-google-search>

<http://infolab.stanford.edu/~backrub/google.html>

Key- Value Store:

http://www.project-voldemort.com/voldemort/design.html

http://s3.amazonaws.com/AllThingsDistributed/sosp/amazon-dynamo-sosp2007.pdf

Consistency in Distributed Systems:

<http://s3.amazonaws.com/AllThingsDistributed/sosp/amazon-dynamo-sosp2007.pdf> - Read 4.4

Cache:

[**https://docs.jivesoftware.com/jive/6.0/community\_admin/index.jsp?topic=/com.jivesoftware.help.sbs.online\_6.0/admin/CachingOverview.html**](https://docs.jivesoftware.com/jive/6.0/community_admin/index.jsp?topic=/com.jivesoftware.help.sbs.online_6.0/admin/CachingOverview.html)

**Design Interview Questions**

You have a potentially very-large set of documents, which are potentially very-large, and contain text. For searching these documents, they've been pre-processed into a (very-large) table mapping words to the set of documents that contain each word. E.g.

(word) : (documents (referenced by ID) containing that word)

Apple: 1, 4, 5, 6, 32

Banana: 5, 6, 7, 9, 32

Cantaloupe: 1, 2, 6

...

Clients will pass in a set of words (e.g. {apple, cantaloupe}), and want the set of document IDs that contain all the words. (e.g. {apple, cantaloupe} -> {1, 6}) Design a distributed system to implement this, bearing in mind that the number of documents, the number of words, and the number of document-IDs-per-word are potentially really, really big

[How would you design a system to provide the top trending topcis in the last 5m/1hour/24hours  The most trending topic should appear first](https://www.careercup.com/question?id=5649012457734144) <https://www.careercup.com/question?id=5649012457734144>

There's requirements from the engineers that they want to limit the right of code files in git server by marking the owner of each file. How would you design the system to solve this problem? What if the user want to use regex when configuring the owner of files?

how to backup between two servers

sliding time window system for big data

2) How would you design the feature in LinkedIn where it computes how many hops there are between you and another person?

Design the services behind an Amazon product page

Recommendation system

Ranking

Distributed logging system - <https://blog.twitter.com/2015/building-distributedlog-twitter-s-high-performance-replicated-log-service>

<http://highscalability.com/product-scribe-facebooks-scalable-logging-system>

**Job Scheduling**

http://www.geeksforgeeks.org/job-sequencing-problem-set-1-greedy-algorithm/

<http://www.geeksforgeeks.org/weighted-job-scheduling/>

<http://www.geeksforgeeks.org/greedy-algorithms-set-1-activity-selection-problem/>

<https://www.careercup.com/question?id=5653760530448384>

<http://massivealgorithms.blogspot.com/2015/08/like-coding-leetcode-253-meeting-rooms.html>

**Stock**

[**http://www.geeksforgeeks.org/stock-buy-sell/**](http://www.geeksforgeeks.org/stock-buy-sell/)

[**https://leetcode.com/problems/best-time-to-buy-and-sell-stock/?tab=Description**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock/?tab=Description)

[**https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/?tab=Solutions**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/?tab=Solutions)

[**https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iii/?tab=Description**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iii/?tab=Description)

[**https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iv/description/**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iv/description/) **AND** [**http://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-k-times**](http://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-k-times)

[**https://discuss.leetcode.com/topic/72011/fb-phone-interview-best-time-to-buy-and-sell-stock-follow-up-with-transaction-fee/14**](https://discuss.leetcode.com/topic/72011/fb-phone-interview-best-time-to-buy-and-sell-stock-follow-up-with-transaction-fee/14)

[**http://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-twice/**](http://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-twice/)

[**https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-cooldown/**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-cooldown/)

**Operations on Large Numbers:**

Multiply infinitely large numbers:

[**http://www.geeksforgeeks.org/multiply-large-numbers-represented-as-strings/**](http://www.geeksforgeeks.org/multiply-large-numbers-represented-as-strings/)

[**http://www.geeksforgeeks.org/divide-large-number-represented-string/**](http://www.geeksforgeeks.org/divide-large-number-represented-string/)

**Points and Distance**

[**http://www.geeksforgeeks.org/find-k-closest-elements-given-value/**](http://www.geeksforgeeks.org/find-k-closest-elements-given-value/)

[**http://www.geeksforgeeks.org/closest-pair-of-points/**](http://www.geeksforgeeks.org/closest-pair-of-points/)

**Tricky:**

**Linklist:**

[**http://quiz.geeksforgeeks.org/delete-occurrences-given-key-linked-list/**](http://quiz.geeksforgeeks.org/delete-occurrences-given-key-linked-list/)

Hashing

Find smallest range containing elements from k lists

Find four elements a, b, c and d in an array such that a+b = c+d

Stack

Next Greater Elem

k stack in array

largest rectangular area

Heap

Tournament Tree

Merge K sorted array

top k freq elements in string/array - implement it

number

find the maximum j – i such that arr[j] > arr[i]

Count smaller elements on right side

Largest subarray with equal number of 0s and 1s

Partiotion problem

first circular tour that visits all petrol pumps

Find subarray with given sum | Set 2 (Handles Negative Numbers)

BST

Inorder predecessor and successor for a given key in BST

kth smallest

Hash

Leet

529

378

421

451

347

216

449

253

STAR method behavioral questions

AVL

REd black

OS

Trie

System design

Gmail

Youtube

Search

Drive

Datacenter

Crack Code

Bit

Stanford

Matrix

Version Number

Job scheduling

meeting room

shortest word distance

bst successor

MS

Longest continuous palindromic substring

|  |
| --- |
| 1. Best time to buy and sell stocks. 2. find the biggest island in a 2D matrix. |

**LeetCode**

**Binary Tree**

94, 145, 144, 637, 235, 404, 110, 1485, 103, 236, 117, 114, 102, 510, 889, 105, 250, 298

**Array**

53, 414, 1314, 1395, 240, 1940, 419, 523(or 560), 15, 56, 432, 1940, 890,

861

Merge K sorted arrays

rotate array

remove duplicates in sorted array

Search an element in a sorted and rotated array

sort by frequency

Maximum difference between 2 such that larger element appears after the smaller

**Linked List**

206, 430, 328, 114,

remove all occurrences of given key from list

sorted link list to BST

Detect and Remove Loop in a Linked List

recurusion - reverse, pairwise swap.

        Reverse a Linked List in groups of given size

        Reverse alternate K nodes in a Singly Linked List

Convert to doubly link list

Flatten Link