

Placement Preparation Resources

We strongly feel that sitting for IT Placements, you must have a good grasp on the topics mentioned below and must also have practised the questions which are listed below. Please do revise (if earlier practised) the questions or do solve them now to prepare for the upcoming IT companies.

The 'expected time to complete' considers 2-3 hours of effort per day.

Sorting

(Expected time to complete : 3-4 days)

The topic we will focus in this section is Sorting. Almost all IT companies test your basics in Coding and Sorting is usually a favourite amongst all IT professionals.

1. Knowing all the basic sorting algorithms, along with their time and space complexity.
2. Also know how to write a custom comparator.

The questions below will give a flavour of different variants of the problem that could be asked from Sorting. The solution for all the problems can be found on geeksforgeeks.com.

Popular Sorting Algorithms:

- a. Bubble Sort
- b. Selection Sort
- c. Insertion Sort
- d. Merge Sort
- e. Quick Sort
- f. Heap Sort
- g. Bucket Sort
- h. Radix Sort

Questions -

1. Custom Comparator

<https://leetcode.com/problems/custom-sort-string/>
<https://leetcode.com/problems/squares-of-a-sorted-array/>
<https://leetcode.com/problems/sort-characters-by-frequency/>
<https://leetcode.com/problems/largest-number/>
<https://leetcode.com/problems/rank-teams-by-votes/>
<https://leetcode.com/problems/filter-restaurants-by-vegan-friendly-price-and-distance/>

2. Intervals

<https://leetcode.com/problems/merge-intervals/>
<https://leetcode.com/problems/meeting-rooms/>
<https://leetcode.com/problems/meeting-rooms-ii/>
<https://leetcode.com/problems/non-overlapping-intervals/>
<https://leetcode.com/problems/insert-interval/>

3. Intersection of two lists

<https://leetcode.com/problems/intersection-of-two-arrays/>
<https://leetcode.com/problems/intersection-of-two-arrays-ii/>

4. Two pointers

<https://leetcode.com/problems/two-sum/>
<https://leetcode.com/problems/3sum/>

5. General Ad-hoc

<https://leetcode.com/problems/longest-word-in-dictionary-through-deleting/>
<https://leetcode.com/problems/divide-array-in-sets-of-k-consecutive-numbers/> <https://leetcode.com/problems/sort-the-matrix-diagonally/>
<https://leetcode.com/problems/pancake-sorting/>

6. Bucket Sort

<https://leetcode.com/problems/longest-word-in-dictionary-through-deleting/>

<https://leetcode.com/problems/divide-array-in-sets-of-k-consecutive-numbers/> <https://leetcode.com/problems/sort-the-matrix-diagonally/>
<https://leetcode.com/problems/pancake-sorting/>

Recursion

(Expected time to complete : 5-7 days)

Reading materials:

<https://www.byte-by-byte.com/recursion/>

<https://www.topcoder.com/community/competitive-programming/tutorials/an-introduction-to-recursion-part-1/>

Framework:

1. Decision at each step
2. Decide subproblems
3. Define the function prototype and mission statement
4. Formulate Recurrence relation
5. Write the base cases
6. Happy implementing!

Questions -

1. Explicit Recursion:

<https://www.hackerrank.com/challenges/recursive-digit-sum/>

<https://leetcode.com/problems/letter-combinations-of-a-phone-number/>

<https://leetcode.com/problems/wildcard-matching/>

<https://leetcode.com/problems/regular-expression-matching/>

2. Subsets alike:

a. Basic to intermediate

<https://leetcode.com/problems/subsets/>

<https://leetcode.com/problems/subsets-ii/>
<https://leetcode.com/problems/combinations/>
<https://leetcode.com/problems/combination-sum/>

b. Intermediate to advanced

<https://leetcode.com/problems/combination-sum-ii/>
<https://leetcode.com/problems/palindrome-partitioning/>
<https://leetcode.com/problems/partition-to-k-equal-sum-subsets/>
<https://leetcode.com/problems/combination-sum-iii/>

3. Arrangement alike:

<https://leetcode.com/problems/permutations/>
<https://leetcode.com/problems/permutations-ii/>
<https://leetcode.com/problems/beautiful-arrangement/>

4. Decisions on Matrices:

<https://leetcode.com/problems/word-search/>
<https://leetcode.com/problems/n-queens/>
<https://leetcode.com/problems/unique-paths-iii/>
<https://leetcode.com/problems/tiling-a-rectangle-with-the-fewest-squares/>
<https://leetcode.com/problems/sudoku-solver/>
<https://leetcode.com/problems/n-queens-ii/>

5. Recursion look-alike mathematical questions:

<https://leetcode.com/problems/permutation-sequence/>
<https://leetcode.com/problems/count-numbers-with-unique-digits/>
<https://leetcode.com/problems/count-all-valid-pickup-and-delivery-options/>

Strings :

(Expected time to complete : 2 days)

<https://leetcode.com/problems/longest-substring-without-repeating-charac>

[ters/](#)

<https://leetcode.com/problems/longest-repeating-character-replacement/>

<https://leetcode.com/problems/minimum-window-substring/>

<https://leetcode.com/problems/valid-anagram/>

<https://leetcode.com/problems/group-anagrams/>

<https://leetcode.com/problems/valid-parentheses/>

<https://leetcode.com/problems/valid-palindrome/>

<https://leetcode.com/problems/longest-palindromic-substring/>

<https://leetcode.com/problems/palindromic-substrings/>

<https://leetcode.com/problems/encode-and-decode-strings/>

Hash Tables

(Expected time to complete : 4-5 days)

<https://leetcode.com/problems/max-points-on-a-line/>

<https://leetcode.com/problems/subarray-sum-equals-k/>

<https://leetcode.com/problems/subarray-sums-divisible-by-k/>

<https://leetcode.com/problems/group-anagrams/>

<https://leetcode.com/problems/fraction-to-recurring-decimal/>

<https://leetcode.com/problems/prison-cells-after-n-days/>

<https://leetcode.com/problems/contiguous-array/>

https://leetcode.com/problems/brick-wall

<https://leetcode.com/problems/rabbits-in-forest/>

Need for ordering:

<https://leetcode.com/problems/minimum-size-subarray-sum/>

Hash tables and 2 pointers:

<https://leetcode.com/problems/subarrays-with-k-different-integers/>

<https://leetcode.com/problems/longest-substring-without-repeating-characters/>

<https://leetcode.com/problems/longest-substring-with-at-most-two-distinct-characters/>

<https://leetcode.com/problems/longest-substring-with-at-most-k-distinct-characters/> <https://leetcode.com/problems/count-number-of-nice-subarrays/>

Search Structures:

(Expected time to complete : 7-8 days)

1. Ad-hoc:

<https://leetcode.com/problems/max-points-on-a-line/>

<https://leetcode.com/problems/group-anagrams/>

<https://leetcode.com/problems/fraction-to-recurring-decimal/> `

2. Subarrays: `

<https://leetcode.com/problems/subarray-sum-equals-k/>

<https://leetcode.com/problems/subarray-sums-divisible-by-k/>

<https://leetcode.com/problems/contiguous-array/> `

3. Frequency counting: `

<https://leetcode.com/problems/brick-wall/>

<https://leetcode.com/problems/rabbits-in-forest/>

`4. Need for ordering: `

<https://leetcode.com/problems/minimum-size-subarray-sum/>

<https://leetcode.com/problems/my-calendar-ii/>

<https://leetcode.com/problems/range-module/>

5. 2 pointers: `

<https://leetcode.com/problems/subarrays-with-k-different-integers/>

<https://leetcode.com/problems/longest-substring-without-repeating-characters/>

<https://leetcode.com/problems/longest-substring-with-at-most-k-distinct-characters/>

<https://leetcode.com/problems/count-number-of-nice-subarrays/>

6. Advance: `

<https://leetcode.com/problems/maximum-equal-frequency/>

Graphs:

(Expected time to complete : 10-12 days)

1. BFS (shortest path):

<https://leetcode.com/problems/word-ladder/>

<https://leetcode.com/problems/01-matrix/>

<https://leetcode.com/problems/as-far-from-land-as-possible/>

<https://leetcode.com/problems/rotting-oranges/>

2. DFS:

<https://leetcode.com/problems/increasing-order-search-tree/>

<https://leetcode.com/problems/critical-connections-in-a-network/>

3. Dijkstra's Algorithm:

<https://leetcode.com/problems/path-with-maximum-minimum-value/>

<https://leetcode.com/problems/path-with-maximum-probability/>

4. Floyd-Warshall/Bellman-ford:

<https://leetcode.com/problems/find-the-city-with-the-smallest-number-of-neighbors-at-a-threshold-distance/>

5. Union Find

<https://leetcode.com/problems/redundant-connection-ii/>

<https://leetcode.com/problems/accounts-merge/>

<https://leetcode.com/problems/satisfiability-of-equality-equations/>

<https://leetcode.com/problems/number-of-operations-to-make-network-connected/>

<https://leetcode.com/problems/most-stones-removed-with-same-row-or-column/>

<https://leetcode.com/problems/number-of-provinces/>

<https://leetcode.com/problems/number-of-provinces/>

6. Graph Colouring

<https://leetcode.com/problems/is-graph-bipartite/>

<https://leetcode.com/problems/possible-bipartition/>

Greedy

(Expected time to complete : 2-3 days)

<https://leetcode.com/problems/minimum-number-of-taps-to-open-to-water-a-garden/>

<https://leetcode.com/problems/maximum-profit-in-job-scheduling/>

<https://leetcode.com/problems/minimum-subsequence-in-non-increasing-order/>

<https://leetcode.com/problems/delete-columns-to-make-sorted/>

<https://leetcode.com/accounts/login/?next=/problems/how-many-apples-can-you-put-into-the-basket/>

Dynamic Programming (DP)

(Expected time to complete : 8-10 days)

1. prefix/suffix DP

<https://leetcode.com/problems/house-robber/>

<https://leetcode.com/problems/maximum-subarray/>

<https://leetcode.com/problems/maximum-product-subarray/>

<https://leetcode.com/problems/longest-turbulent-subarray/>

<https://leetcode.com/problems/longest-valid-parentheses/>

<https://leetcode.com/problems/climbing-stairs/>

<https://leetcode.com/problems/decode-ways/>

<https://leetcode.com/problems/integer-break/>

2. Matrix DP

<https://leetcode.com/problems/minimum-path-sum/>

<https://leetcode.com/problems/count-square-submatrices-with-all-ones/>

<https://www.interviewbit.com/problems/min-sum-path-in-matrix/>

<https://www.interviewbit.com/problems/dungeon-princess/>

<https://leetcode.com/problems/cherry-pickup/>

<https://www.interviewbit.com/problems/kingdom-war/>
<https://www.interviewbit.com/problems/unique-paths-in-a-grid/>

3. Tree DP:

<https://leetcode.com/problems/maximum-sum-bst-in-binary-tree/>
<https://leetcode.com/problems/longest-zigzag-path-in-a-binary-tree/>

4. Digit DP:

<https://leetcode.com/problems/count-numbers-with-unique-digits/>
<https://leetcode.com/problems/numbers-at-most-n-given-digit-set/>
<https://leetcode.com/problems/numbers-with-repeated-digits/>
<https://leetcode.com/problems/form-largest-integer-with-digits-that-add-up-to-target/>

5. DP with bitmasking

<https://leetcode.com/problems/find-the-shortest-superstring/>
<https://leetcode.com/problems/partition-to-k-equal-sum-subsets/>
<https://leetcode.com/problems/maximum-students-taking-exam/>
<https://leetcode.com/problems/number-of-islands/>

Connectivity and reachability

(Expected time to complete : 4-5 days)

<https://leetcode.com/problems/number-of-islands/>
<https://leetcode.com/problems/number-of-distinct-islands/>
<https://leetcode.com/problems/surrounded-regions/>
<https://leetcode.com/problems/number-of-enclaves/>
<https://leetcode.com/problems/number-of-closed-islands/>
<https://leetcode.com/problems/keys-and-rooms/>
<https://leetcode.com/problems/max-area-of-island/>
<https://leetcode.com/problems/flood-fill/>
<https://leetcode.com/problems/jump-game-iii/>
<https://leetcode.com/problems/course-schedule-iv/>

DS Design

(Expected time to complete : 5-7 days)

<https://leetcode.com/problems/min-stack/>

<https://leetcode.com/problems/time-based-key-value-store/>

<https://leetcode.com/problems/peeking-iterator/>

<https://www.interviewbit.com/problems/sliding-window-maximum/>

<https://leetcode.com/problems/copy-list-with-random-pointer/>

<https://leetcode.com/problems/maximum-frequency-stack/>

<https://leetcode.com/problems/maximum-equal-frequency/>

<https://leetcode.com/problems/insert-delete-getrandom-o1/>

<https://leetcode.com/problems/insert-delete-getrandom-o1-duplicates-allowed/>

<https://leetcode.com/problems/all-ones-data-structure/>

<https://leetcode.com/problems/lru-cache/>

<https://leetcode.com/problems/sliding-window-median/>

Tree

(Expected time to complete : 12-15 days)

1. Tree Traversal -

<https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal/>

<https://leetcode.com/problems/binary-tree-level-order-traversal-ii/>

<https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal/>

<https://leetcode.com/problems/construct-binary-tree-from-inorder-and-postorder-traversal/>

<https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal/>

2. Tree Structure -

<https://leetcode.com/problems/symmetric-tree/>

<https://leetcode.com/problems/invert-binary-tree/>

3. Path Sum -

<https://leetcode.com/problems/path-sum-ii/>

<https://leetcode.com/problems/binary-tree-maximum-path-sum/>
<https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree/>

4. Miscellaneous

<https://leetcode.com/problems/maximum-depth-of-binary-tree/>
<https://leetcode.com/problems/same-tree/>
<https://leetcode.com/problems/invert-binary-tree/>
<https://leetcode.com/problems/binary-tree-maximum-path-sum/>
<https://leetcode.com/problems/binary-tree-level-order-traversal/>
<https://leetcode.com/problems/serialize-and-deserialize-binarytree/>
<https://leetcode.com/problems/subtree-of-another-tree/>
<https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal/>
<https://leetcode.com/problems/validate-binary-search-tree/>
<https://leetcode.com/problems/kth-smallest-element-in-a-bst/>
<https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-search-tree/>
<https://leetcode.com/problems/add-and-search-word-data-structure-design/>

Trie

(Expected time to complete : 2-4 days)

<https://leetcode.com/problems/implement-trie-prefix-tree/>
<https://leetcode.com/problems/add-and-search-word-data-structure-design/>
<https://leetcode.com/problems/concatenated-words/>
<https://leetcode.com/problems/design-search-autocomplete-system/>
<https://leetcode.com/problems/word-search-ii/>

Reading Material on Object Oriented Programming (OOP)

C:

<https://medium.com/@somiltgupta/how-to-learn-c-via-only-i-o-questions-3f4be1e30365>

JAVA:

<https://medium.com/@somiltgupta/oop-java-reading-references-part-2-oops-concepts-c146058c0fd5>

<https://medium.com/@somiltgupta/oop-java-reading-references-part-2-oops-concepts-c146058c0fd5>

<https://medium.com/@somiltgupta/oop-java-reading-references-part-3-java-constructs-basics-d7e7882c7937>