Competitive Coding

Seasons of Code, 2022

HARSH POONIA

July 2022

Overview

1. C++ Standard Template Library
2. Sorting and Searching
3. Graph Algorithms
4. Divide and Conquer
5. Dynamic Programming
6. Greedy Algorithms

C++ STL

The Standard Template Library provides with tools to represent data in ways that allow us to store, retreive and process that data efficiently.

Some of the useful data structures in STL are

- **Vectors** dynamic, contiguous memory location
- **Strings** dynamic, similar to character arrays
- **Queue** Queue is First In First Out (FIFO) data type
- **Deque** A double sided queue.
- **Stack** A Last In First Out (LIFO) data type, used for recursive and functional calls
- **Pair, Sets, Maps, Multisets** Maps create a mapping from data type to another, with a key-value relationship

Sorting and Searching

Binary Search

Binary Search is the technique of searching for an element in a sorted array, which involves repeatedly reducing our search area by a factor of 2, and thus the time complexity is $\mathcal{O}(log_2(n))$.
Binary search eleiminates the need to scan every element of the array, which would take linear ($\mathcal{O}(n)$) time.

Some Sorting Algorithms

Insertion sort Inserts new elements into their apt place in an already sorted array

Merge sort Splits the array to be sorted into 2, sorts them separately, and merges the sorted subarrays, $\mathcal{O}(nlog(n))$

Counting sort Maintains a count of element occurrences, $\mathcal{O}(n)$ time complexity, *only for arrays with largest element* $\approx n$

Graphs

Graph A graph is a non-linear data structure containing nodes and edges. Nodes are also refered to as vertices, and edges connect a pair of nodes.

Path A path from vertex u to v is a sequence of vertices v_i such that consecutive vertices are adjacent.

Cycle A path such that it starts and ends at the same node.

Connected graphs A graph is connected if there is a path from any node to any other node, ie there exists a path between any pair of distinct vertices.

Directed graphs edges have a direction, ie edges are an ordered pair {v, w}

Tree A tree is an undirected graph in which any pair of vertices are connected by exactly one path. Or equivalently,

Tree

A connected, acyclic, undirected graph is called a tree.

Algorithms for graphs

Traversal

Depth First Search DFS is an algorithm in which we start at a root node, and explore as far as possible along a branch before backtracking. Because we go to full depth of a branch and then come back, hence the name *depth-first*.

Breadth First Search BFS is an algorithm where we start at a root node and we explore all the possible nodes at a depth before moving on to nodes at the next depth level. Here we are covering entire breadth of the graph at some level/distance from root node.

Divide and Conquer

A programming paradigm that is incredibly useful in solving problems where the optimal solution to a problem can be expressed in terms of solutions to its sub-problems. We essentially divided the larger problem and tackle the smaller problems.

- In the tiling trominos problem, we split the board (of side length 2^{k+1}) into 4 pieces, each of side length 2^k, and use solutions to these 4 parts.
- To find the closest pair of points in a given set of points in the 2-D plane, we can divide the plane into 2 parts, find the closest distances separately in those 2 parts, and use some constraints to find cross pairs that might give smaller distance.

Dynamic Programming

The basic motive behind dynamic programming is to solve reccurence relations faster. We wish to eliminate redundant recursive calls, and thus avoid repeat calculations. We store already computed values and use them.

Memoisation is a technique that stores already calculated recursive solutions in a memo, and fetch values from the memo everytime the function is called again. It is thus a *top-down approach*.

Tabulation We start by solving the subproblems, smallest first. Thus for each subproblem, we wold already have solved all its previous subproblems, effectively solving this subproblem. Thus it is a *bottom-up approach.*

Greedy Algorithms

- Greedy algorithms try to find locally optimal solutions that lead to globally optimal solutions.
- Like DP, greedy algorithms also exploit the optimal substructure of a problem.
- But, we need to ensure that locally optimal solutions (greedy choice)at each step leads to a globally optimal solution. Only then we proceed to devise a greedy algorithm.