

Data Structures and Algorithms

Prajval

Contents

1. Big O	2
2. Data Structures	2
2.1 Arrays	2
3. Algorithms	3
3.1 Search	3
3.1.1 Linear Search	3
3.1.2 Binary Search	3
3.2 Sorting	3
3.2.1 Bubble Sort	3
3.2.2 Selection Sort	3
3.3 Problems	3
3.3.1 Two crystal ball problem	3
Implementation	3

1. Big O

- growth is w.r.t. the input
- constants are dropped
- worst case is usually the way we measure

2. Data Structures

2.1 Arrays

- contiguous memory space
- fixed size
- all operations in array take $O(1)$, time taken for each cannot grow when input grows
- $a[n] = \text{address of } a + \text{offset}(n) * \text{width of type}$

3. Algorithms

3.1 Search

3.1.1 Linear Search

- go through each value in the array and check if its the required value
- $O(N)$, grows linearly with input

3.1.2 Binary Search

- array should be ordered
- divide the array and check which half the key is present.
- $O(\log N)$

3.2 Sorting

- array is sorted if any $x[i] \leq x[i + 1]$

3.2.1 Bubble Sort

- iterate over the array and check if $x[i] > x[i + 1]$ then swap each other
- for each iteration the largest item is sorted
- $N(N+1)/2 = N^2 + N \Rightarrow O(N^2)$ // here N is insignificant

3.2.2 Selection Sort

- Selection Sort is a simple sorting algorithm that repeatedly finds the minimum element from the unsorted part of the array and places it at the beginning.

3.3 Problems

3.3.1 Two crystall ball problem

given two crystall balls that will break if dropped from high enough distance, determine the exact spot in which it will break in the most optimized way.

Implementation

<https://github.com/prajwal-badiger/dsa.git>