Lecture 08

Algorithms and Data structures

Algorithms

Algorithms derives from name of Central Asian scientist Al-Khorezmi.

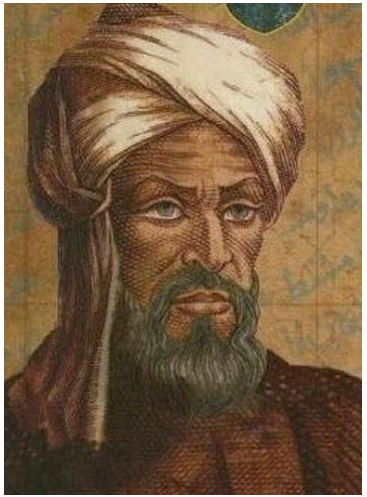
Algorithm is a step-by-step set of operations to solve some problem

Major Computer Science problems that are needed to solve are:

- Searching
- Sorting

There can be many algorithms to solve on problem, such as:

- Quick sort
- Bubble sort



Searching

Linear search

In plain English, Linear Search algorithm is as follows:

- Check if the first item in a list is the item you are searching for, if it is the one you are looking for, you are done.
- If it isn't the item you are searching for move on and check the next item.
- Continue checking items until you find the one you are searching for.

Binary search

Binary Search is a very powerful algorithm. If you had **1000** presents to search through it would take you at most **10** checks for Binary search to find something and Linear search would take at most 1000 checks, but if you doubled the number of presents to search through how would this change the number of checks made by Binary Search and Linear search?

Binary Search (pseudo code)

- Look at the item in the centre of the list and compare it to what you are searching for
- If it is what you are looking for then you are done.
- If it is larger than the item you are looking for then you can ignore all the items in the list which are larger than that item (if the list is from smallest to largest this means you can ignore all the items to the right of the centre item).
- If it is smaller then you can ignore all the items in the list which are smaller than that centre item.
- Now repeat the algorithm on the remaining half of the list, checking the middle of the list and choosing one of the halves, until you find the item you are searching for.

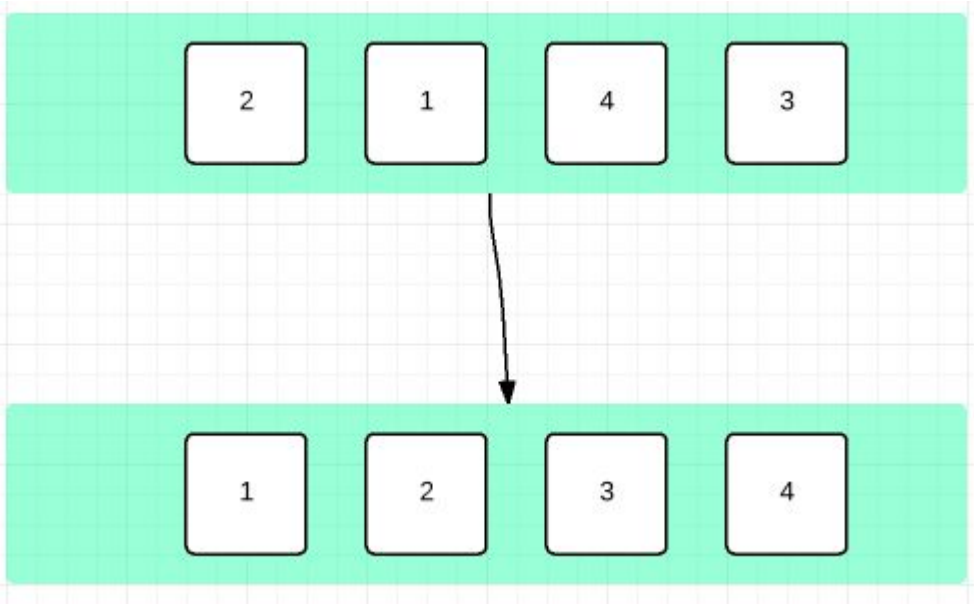
Binary Search vs Linear Search

We have 1000 elements

Position of element	**Binary Search**	**Linear Search**
In the beginning	10 operations	1 operations
In the center	1 operation	500 operations
In the end	10 operations	1000 operations
In 250th or 750th	2 operations	250 or 750 operations

Sorting

Very important area of computer programming



Bubble sort

Selection sort

The selection sort algorithm can be described as follows:

- Find the smallest item in the list and place it to one side. This will be your sorted list.
- Next find the smallest item in the remaining list, remove it and place it into your sorted list beside the item you previously put to the side.
- Repeat this process until all items have been selected and moved into their correct position in the sorted list.

Insertion sort

Insertion sort can be described with informal instructions as follows:

- Take an item from your unsorted list and place it to the side, this will be your sorted list.
- One by one, take each item from the unsorted list and insert it into the correct position in the sorted list.
- Do this until all items have been sorted.

Quicksort

Quicksort can be described in the following way:

- Choose an item from the list and compare every other item in the list to this (this item is often called the pivot).
- Place all the items that are greater than it into one subgroup and all the items that are smaller into another subgroup. Place the pivot item in between these two subgroups.
- Choose a subgroup and repeat this process. Eventually each subgroup will contain only one item and at this stage the items will be in sorted order.

Time and Space complexity

Every algorithm uses some space and spends some time to solve problem

Best, worst and average

Any algorithm has its own best, worst and average case

For example:

linear search - best case 1, worst case n

binary search - best case 1, worst case log(n). why?

Data structures

We need to store some data, so that it can be:

- fastly find needed element
- fastly remove needed element
- save them in specific order

Array or list

Simplest data structure, saving it in memory sequence

Finding n-th element is done in 1 operation

Inserting one element is done in n-k operations

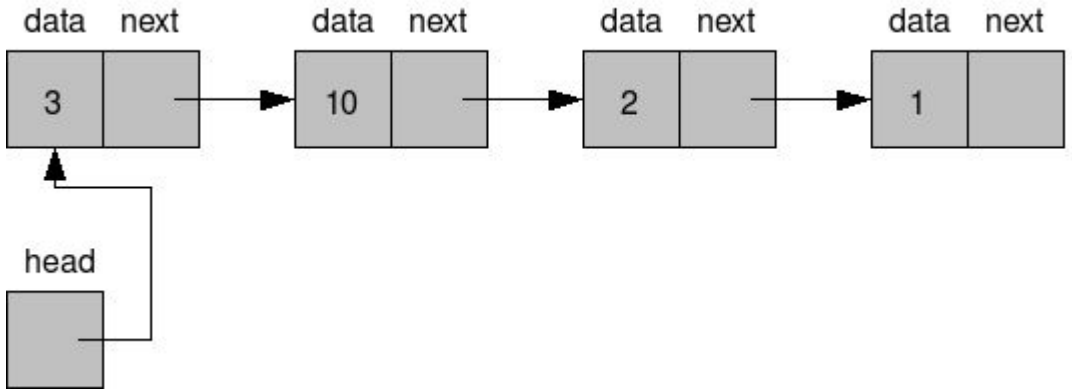
Deleting is done in n-operations

LinkedList

LinkedList each element is connected with next element by link

So finding n-th element takes n operations

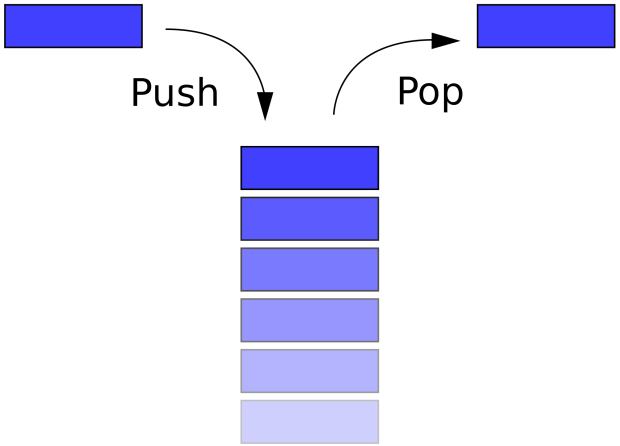
Inserting in any place takes 1 operation

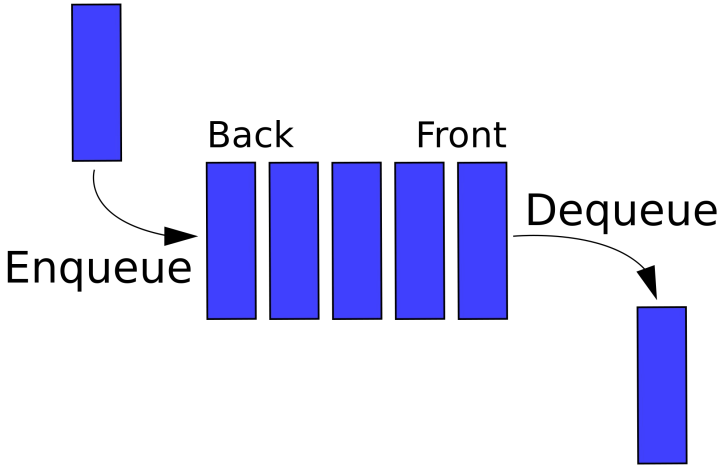


HashSet

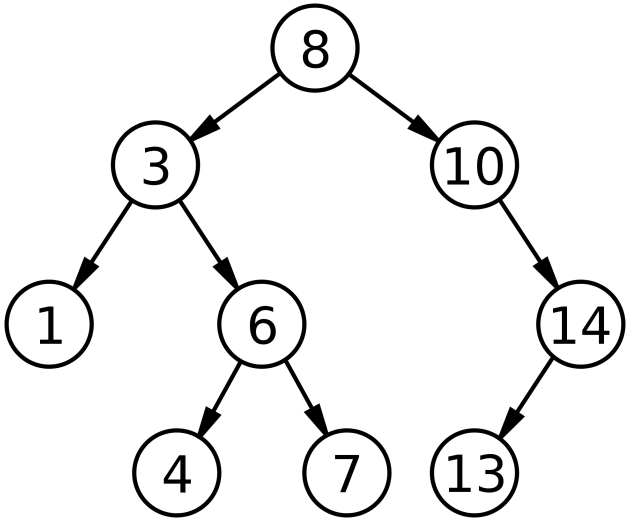
Searching by Hash takes 1 operation

Stack, Queue

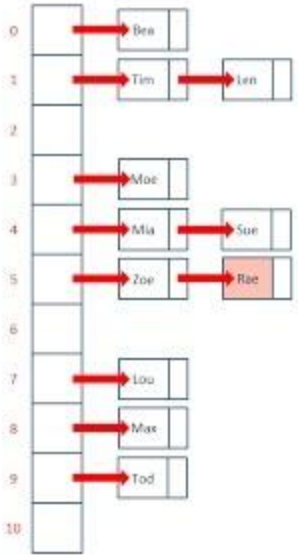




Trees, Black-White trees



HashSet and TreeSet



Find Rae 280 Mod 11 = 5

myData = Array(5)