VTC Academy
Think Ahead, Beyond Boundaries

DATA STRUCTURES AND ALGORITHMS

LESSON 5

Sorting and Searching Algorithms

CONTENT

VTC Academy
Think Ahead, Beyond Boundaries

1. Sorting Algorithms
    - Bubble Sort
    - Quick Sort
    - Heap Sort
2. Searching Algorithms
    - Linear Search
    - Binary Search
3. Summary

Sorting Algorithms

VTC Academy
Think Ahead, Beyond Boundaries

- Sorting refers to arranging data in a particular format
- Sorting algorithm specifies the way to arrange data in a particular order
- Most common orders are in numerical or lexicographical order
- The importance of sorting lies in the fact that data searching can be optimized to a very high level, if data is stored in a sorted manner
- Sorting is also used to represent data in more readable formats.

Types of Sorting Algorithm

VTC Academy
Think Ahead, Beyond Boundaries

- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Heap Sort

Important Terms

VTC Academy
Think Ahead, Beyond Boundaries

- Increasing Order
  - A sequence of values is said to be in increasing order, if the successive element is greater than the previous one.
  - For example: 1, 3, 4, 6, 8, 9 are in increasing order, as every next element is greater than the previous element.
- Decreasing Order
  - A sequence of values is said to be in decreasing order, if the successive element is less than the current one.
  - For example, 9, 8, 6, 4, 3, 1 are in decreasing order, as every next element is less than the previous element.

Important Terms

VTC Academy
Think Ahead, Beyond Boundaries

- Non-Increasing Order
  - A sequence of values is said to be in non-increasing order, if the successive element is less than or equal to its previous element in the sequence. This order occurs when the sequence contains duplicate values.
  - For example, 9, 8, 6, 3, 3, 1 are in non-increasing order, as every next element is less than or equal to (in case of 3) but not greater than any previous element.

Important Terms

VTC Academy
Think Ahead, Beyond Boundaries

- Non-Increasing Order
  - A sequence of values is said to be in non-increasing order, if the successive element is less than or equal to its previous element in the sequence. This order occurs when the sequence contains duplicate values.
  - For example, 9, 8, 6, 3, 3, 1 are in non-increasing order, as every next element is less than or equal to (in case of 3) but not greater than any previous element.

Bubble Sort

VTC Academy
Think Ahead, Beyond Boundaries

- Bubble sort is a simple sorting algorithm.
- Bubble sort is a sorting algorithm that compares two adjacent elements and swaps them until they are not in the intended order.
- Just like the movement of air bubbles in the water that rise up to the surface, each element of the array move to the end in each iteration. Therefore, it is called a bubble sort.

6 5 3 1 8 7 2 4

Source: Wikipedia.

Optimized Bubble Sort

VTC Academy
Think Ahead, Beyond Boundaries

- In the basic bubble sort algorithm, all the comparisons are made even if the array is already sorted. This increases the execution time.
- To solve this, we can introduce an extra variable swapped. The value of swapped is set true if there occurs swapping of elements. Otherwise, it is set false.
- After an iteration, if there is no swapping, the value of swapped will be false. This means elements are already sorted and there is no need to perform further iterations.
- This will reduce the execution time and helps to optimize the bubble sort.

Optimized Bubble Sort

VTC Academy
Think Ahead, Beyond Boundaries

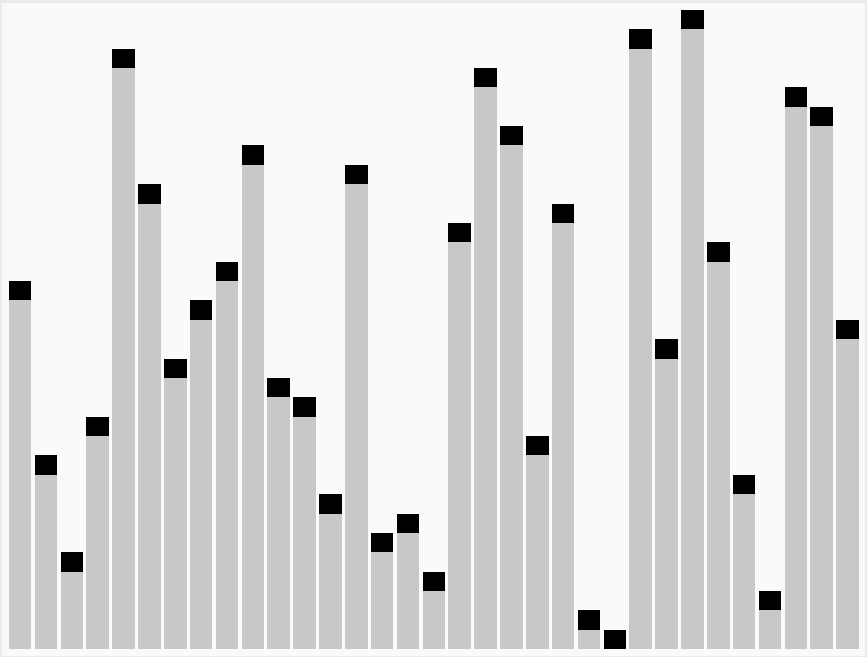
```
void bubbleSort(int arr[], int n) {
    int i, j;
    int swapped;
    for (i = 0; i < n-1; i++) {
        swapped = 0;
        for (j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                swap(&arr[j], &arr[j+1]);
                swapped = true;
            }
        }
        if (swapped == false) {
            break;
        }
    }
}
```

Quick Sort

VTC Academy
Think Ahead, Beyond Boundaries

- Quick sort is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller arrays.
- A large array is partitioned into two arrays one of which holds values smaller than the specified value, say pivot, based on which the partition is made and another array holds values greater than the pivot value.
- Quick sort partitions an array and then calls itself recursively twice to sort the two resulting subarrays.
- This algorithm is quite efficient for large-sized data sets as its average and worst case complexity are of O(n2), where n is the number of items.

Quick Sort



Animated visualization of the quicksort algorithm. The horizontal lines are pivot values. Source: Wikipedia.

Quick Sort

VTC Academy
Think Ahead, Beyond Boundaries

- Based on our understanding of partitioning in quick sort, we will now try to write an algorithm for it, which is as follows:
  - Step 1 – Choose the highest index value has pivot
  - Step 2 – Take two variables to point left and right of the list excluding pivot
  - Step 3 – left points to the low index
  - Step 4 – right points to the high
  - Step 5 – while value at left is less than pivot move right
  - Step 6 – while value at right is greater than pivot move left
  - Step 7 – if both step 5 and step 6 does not match swap left and right
  - Step 8 – if left ≥ right, the point where they met is new pivot

Quick Sort Algorithm

VTC Academy
Think Ahead, Beyond Boundaries

- Using pivot algorithm recursively, we end up with smaller possible partitions. Each partition is then processed for quick sort.
  - Step 1 – Make the right-most index value pivot
  - Step 2 – Partition the array using pivot value
  - Step 3 – Quicksort left partition recursively
  - Step 4 – Quicksort right partition recursively

Quick Sort Algorithm

VTC Academy
Think Ahead, Beyond Boundaries

```
int partition(int *a, int left, int right, int pivot){
    int leftIndex = left - 1;
    int rightIndex = right;
    int temp;
    while(1){
        while(a[++leftIndex]< pivot);
        while(rightIndex>0 && a[--rightIndex] > pivot);
        if(leftIndex >= rightIndex) {
            break;
        }else{
            temp = a[leftIndex];
            a[leftIndex] = a[rightIndex];
            a[rightIndex] = temp;
        }
    }
    return leftIndex;
}
```

Quick Sort Algorithm

VTC Academy
Think Ahead, Beyond Boundaries

```
void quickSort(int* a, int left, int right){
    if(right<left){
        return;
    }
    int pivot = a[right];
    int partitionIndex = partition(a, left, right, pivot);

    a[right] = a[partitionIndex];
    a[partitionIndex] = pivot;

    quickSort(a, left, partitionIndex-1);
    quickSort(a, partitionIndex+1, right);
}
```

Heap Sort

VTC Academy
Think Ahead, Beyond Boundaries

- In computer science, heapsort is a comparison-based sorting algorithm.
- Heapsort can be thought of as an improved selection sort: like selection sort, heapsort divides its input into a sorted and an unsorted region, and it iteratively shrinks the unsorted region by extracting the largest element from it and inserting it into the sorted region.
- Unlike selection sort, heapsort does not waste time with a linear-time scan of the unsorted region; rather, heap sort maintains the unsorted region in a heap data structure to more quickly find the largest element in each step.

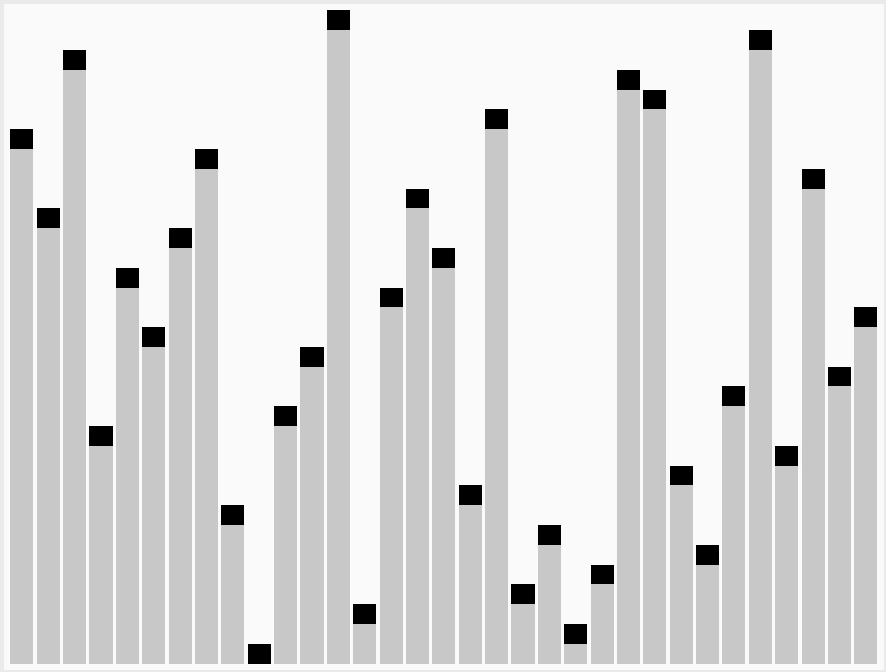
Heap Sort

VTC Academy
Think Ahead, Beyond Boundaries

- Heap Sort is a popular and efficient sorting algorithm in computer programming. Learning how to write the heap sort algorithm requires knowledge of two types of data structures - arrays and trees.
- Heap sort works by visualizing the elements of the array as a special kind of complete binary tree called a heap.
- As a prerequisite, you must know about a complete binary tree and heap data structure.

Heap Sort

- A run of heapsort sorting an array of randomly permuted values.
- In the first stage of the algorithm the array elements are reordered to satisfy the heap property.
- Before the actual sorting takes place, the heap tree structure is shown briefly for illustration.



Source: Wikipedia.

What is Heap Data Structure?

VTC Academy
Think Ahead, Beyond Boundaries

- Heap is a special tree-based data structure.
- A binary tree is said to follow a heap data structure if
  - It is a complete binary tree.
  - All nodes in the tree follow the property that they are greater than their children i.e. the largest element is at the root and both its children and smaller than the root and so on. Such a heap is called a max-heap. If instead, all nodes are smaller than their children, it is called a min-heap.

What is Heap Data Structure?

VTC Academy
Think Ahead, Beyond Boundaries

- Min-Heap – Where the value of the root node is less than or equal to either of its children.
- Max-Heap – Where the value of the root node is greater than or equal to either of its children.

How to “Heapify” a Tree?

VTC Academy
Think Ahead, Beyond Boundaries

- The process of reshaping a binary tree into a Heap data structure is known as ‘heapify’. A binary tree is a tree data structure that has two child nodes at max. If a node’s children nodes are ‘heapified’, then only ‘heapify’ process can be applied over that node.
- Starting from a complete binary tree, we can modify it to become a Max-Heap by running a function called ‘heapify’ on all the non-leaf elements of the heap. i.e. ‘heapify’ uses recursion.

How to “Heapify” a Tree?

VTC Academy
Think Ahead, Beyond Boundaries

- Example of “heapify”:

```
          30(0)
         /    \
     70(1)    50(2)

Child (70(1)) is greater than the parent (30(0))

Swap Child (70(1)) with the parent (30(0))
          70(0)
         /    \
     30(1)    50(2)
```

Working of Heap Sort

VTC Academy
Think Ahead, Beyond Boundaries

1. Since the tree satisfies Max-Heap property, then the largest item is stored at the root node.
2. Swap: Remove the root element and put at the end of the array (nth position) Put the last item of the tree (heap) at the vacant place.
3. Remove: Reduce the size of the heap by 1.
4. Heapify: Heapify the root element again so that we have the highest element at root.
5. The process is repeated until all the items of the list are sorted.

Working of Heap Sort

VTC Academy
Think Ahead, Beyond Boundaries

```
void heapify(int arr[], int n, int i) {
    int largest = i;
    int left = 2 * i + 1;
    int right = 2 * i + 2;
    if (left < n && arr[left] > arr[largest])
        largest = left;
    if (right < n && arr[right] > arr[largest])
        largest = right;
    if (largest != i) {
        swap(&arr[i], &arr[largest]);
        heapify(arr, n, largest);
    }
}
```

Working of Heap Sort

VTC Academy
Think Ahead, Beyond Boundaries

```
void heapSort(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--) {
        heapify(arr, n, i);
    }
    for (int i = n - 1; i >= 0; i--) {
        swap(&arr[0], &arr[i]);
        heapify(arr, i, 0);
    }
}
```

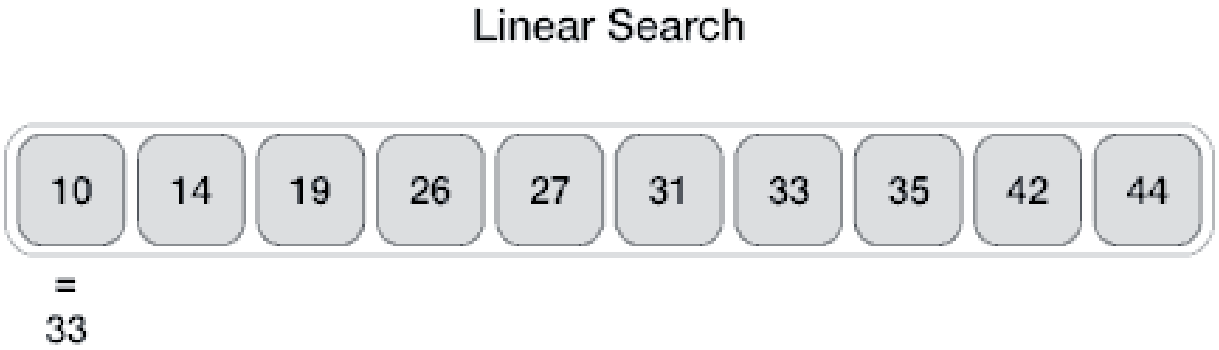
Searching Algorithms

VTC Academy
Think Ahead, Beyond Boundaries

- In computer science, a search algorithm is an algorithm which solves a search problem. Search algorithms work to retrieve information stored within some data structure, or calculated in the search space of a problem domain, either with discrete or continuous values.
- While the search problems described above and web search are both problems in information retrieval, they are generally studied as separate subfields and are solved and evaluated differently. Web search problems are generally focused on filtering and finding documents that are most relevant to human queries.

Linear Search

- Linear search is a very simple search algorithm. In this type of search, a sequential search is made over all items one by one
- Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.



Linear Search

VTC Academy
Think Ahead, Beyond Boundaries

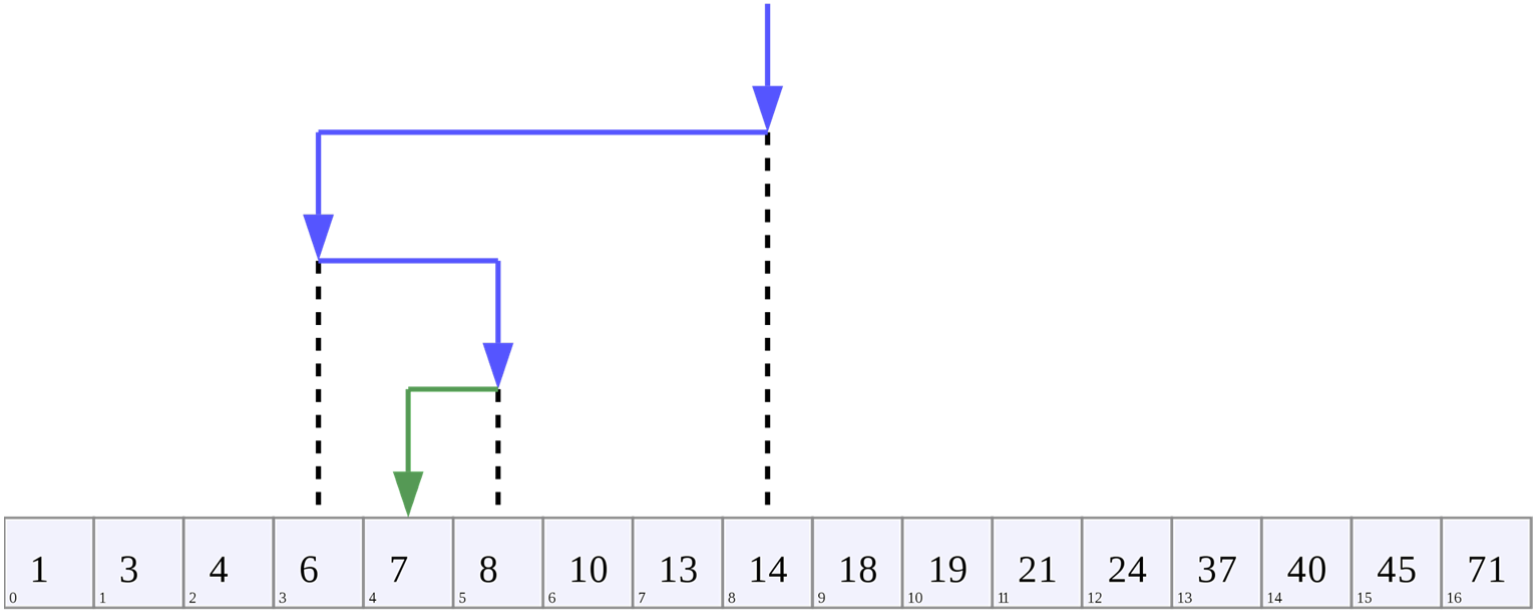
```
int linearSearch(int* data, int dataSize, int searchValue){
    for(int i=0; i<dataSize; i++){
        if(data[i]==searchValue){
            return i;
        }
    }
    return -1;
}
```

Binary Search

VTC Academy
Think Ahead, Beyond Boundaries

- Binary search is a fast search algorithm
- Binary Search is a searching algorithm for finding an element's position in a sorted array.
- In this approach, the element is always searched in the middle of a portion of an array.
- Binary search can be implemented only on a sorted list of items. If the elements are not sorted already, we need to sort them first.

Binary Search



Visualization of the binary search algorithm where 7 is the target value.
Source: Wikipedia.

Binary Search Working

VTC Academy
Think Ahead, Beyond Boundaries

- Binary Search Algorithm can be implemented in two ways:
  - Iterative Method
  - Recursive Method
- The recursive method follows the divide and conquer approach.

Binary Search – Iterative Method

VTC Academy
Think Ahead, Beyond Boundaries

```
int binarySearch(int arr[], int l, int r, int x) {
    while (l <= r) {
        int m = l + (r - l) / 2;
        if (arr[m] == x) {
            return m;
        }
        if (arr[m] < x) {
            l = m + 1;
        }
        else {
            r = m - 1;
        }
    }
    return -1;
}
```

Binary Search – Recursive Method

VTC Academy
Think Ahead, Beyond Boundaries

```
int binarySearch(int arr[], int l, int r, int x) {
    if (r >= l) {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x) {
            return mid;
        }
        if (arr[mid] > x) {
            return binarySearch(arr, l, mid - 1, x);
        }
        return binarySearch(arr, mid + 1, r, x);
    }
    return -1;
}
```

Summary

VTC Academy
Think Ahead, Beyond Boundaries

- Bubble sort algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.
- Quick sort is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller arrays. A large array is partitioned into two arrays one of which holds values smaller than the specified value, say pivot, based on which the partition is made and another array holds values greater than the pivot value.

Summary

VTC Academy
Think Ahead, Beyond Boundaries

- Heap Sort is a popular and efficient sorting algorithm in computer programming. Learning how to write the heap sort algorithm requires knowledge of two types of data structures - arrays and trees.
- Linear search is a very simple search algorithm. In this type of search, a sequential search is made over all items one by one.
- Binary search is a fast search algorithm with run-time complexity of O(log n). This search algorithm works on the principle of divide and conquer.

VTC Academy
Think Ahead, Beyond Boundaries