Searching Algorithms

Tutorial 4

1. Using an appropriate **array** example, explain how main **searching** algorithms can be performed.
2. Compare (look for similarities) and contrast (look for differences) **linear** search and **binary** search algorithms.
3. Write a function using pseudo or source codes for searching an integer variable called an ***item*** using **linear search** in an **array** called an ***unordered array***.

#include<stdio.h>

#include <stdbool.h>

int main(){

int item = 8;

int unordered\_Array[] = {5, 8, 6, 9, 10, 43, 56, 76, 2, 38};

int array\_length = sizeof(unordered\_Array);

int position = -1;

bool found = false;

int i =0;

while(i < array\_length && found == false){

if (item == unordered\_Array[i]){

found = true;

position = i;

printf("item found at %d ", position);

}

i++;

}

return 0;

}

1. Write the C program for **Binary** Search.

int binary\_search(int arr[], int Search\_Element, int array\_length){

int first=0;

int last=N-1;

bool found= false;

int position=-1;

while(first<=last && !found){

int mid= (first+last)/2;

if(Search\_Element==arr[mid]){

found=true;

position=mid;

}else if(arr[mid]>Search\_Element){

//search at the lower boundary

last=mid-1;

}else{

first=mid+1;

}

}

return position;

}

1. Briefly Explain Bubble Sort, Selection Sort, and Insertion Sort