Tutorials (4)

04/5/2022

① Using an appropriate array example, explain how main searching algorithms can be performed

There are two main searching algorithms. They are linear search and binary search.

For example; In the following sorted set of numbers, the linear and binary searches can be mentioned as follows.

2, 4, 6, 8, 9, 10, 12.

If the key value is 8,

Linear search

Here, we search the key value 8 by searching one by one in a sequential manner.

Binary search

Usually an unsorted set of numbers should be rearranged in an order before the binary search starts. Then, Considering the array length, the mid value is considered then, if the key value is inside any side of the mid value only that side is going to be considered in the next steps until the key value is isolated.

② Compare and contrast linear search and binary search algorithms.

Done in previous tutorial

⑧ Write an implementation for both linear and binary searches using c language programme. Use a variable called 'item' in an unordered array

```
#include <stdio.h>
#include <stdlib.h>
main()
{
/* Declare variables - array of number, search key, i, j, low, high */

int array[100], search_key, i, n, low, high, location, choice;
void linear_search (int search_key, int array[100], int n);
void binary_search (int search_key, int array[100], int n);

Clrscr();
/* Read the elements of array */
printf ("Enter the size of the array:");
Scanf ("%d", &n);
Printf ("Enter the elements of the array: \n");
for (i=1; i<=n; i++)
{
    scanf ("%d", &array[i]);
}
/* Get the search key element for Linear search */
Printf ("Enter the search key: ");
Scanf ("%d", &search_key);

/* Choice of Search Algorithm */
printf ("_____\n");
printf ("1. Linear search \n");
printf ("2. Binary search \n");
printf ("_____\n");
printf ("Enter your choice");
Scanf ("%d", &choice);
```

```
switch (choice)
{
   case 1;
   linear_search (search_key, array, n);

   break;

   case 2;
   binary_search (search_key, array, n);

   break;

  default;
   exit (0);
}
   getch();
   return 0;
}
 /* Declare variable */

  int i, location;
for (i=1; i<=n; i++)
{
    if (search_key == array[i])
     {
          location = i;
printf ("_____\n");
printf ("The location of search key = %d is %d \n",
                                search_key, location);
printf ("_____\n");
}
}
}
```

```
/* Binary Search to find Search Key */

Void binary_search (int search_key, int array [100], int n)
{
    int mid, i, low, high;
    low = 1;
    high = n;
    mid = (low + high) / 2;

  1 = 1;
    while (search_key != array [mid])
  {
    if (search_key <= array [mid])
    {
        low = 1;
        high = mid + 1;
        mid = (low + high) / 2;
    }
    else
    {
        low = mid + 1;
        high = n
        mid = (low + high) / 2;
    }

  }
printf ("_____ \n");
printf ("location = %d \t", mid);
printf ("Search_key = %d found! \n", search_key);
printf ("_____ \n");
}
```