

Data structures and Algorithms

Using "C".

S. Srikar

APIA110010474

Assignment - 6.

CSE - F

(1)
(a)

```
#include <stdio.h>

int binary_search (int arr[], int s, int r, int, i)
{
    if (r == s)
    {
        int mid =  $s + (r - s) / 2$ 
        if (arr[mid] == i)
            return mid;
        if (arr[mid] > i)
            return binary_search (arr, s, mid - 1, i);
        return binary_search (arr, mid + 1, r, i);
    }
    return -1;
}

int main ()
{
    int num;
    printf ("Hello brother enter any size:");
    scanf ("%d", &num);
}
```

Srikan

```
int k, a, var[num], op, var, p1, p2, sum, pr;
```

```
for (i=0; i<num; i++)
```

```
{
```

```
    printf("Hey enter value");
```

```
    scanf("%d", &var[i]);
```

```
}
```

```
for (k=0; k<num; k++)
```

```
{
```

```
if (var[k] < var[k+1])
```

```
for (b=k+1; b<num; b++)
```

```
    if (var[k] < var[b])
```

```
    {
```

```
        a = var[k];
```

```
        var[k] = var[b];
```

```
        var[b] = a;
```

```
    }
```

```
} printf("enter array in decreasing order list");
```

```
for (k=0; k<num; k++)
```

```
{
```

```
    printf("%d", var[k]);
```

```
}
```

```
printf("\n * MENU * \n");
```

```
printf("i. Enter value position finding:");
```

```
printf("ii. Find position Now:");
```

```
printf("iii. Print sum and Product);
```


[Srikanth API alluwa] 474
Print ("In Please enter your Preference");

scanf ("%d", &op);
switch (op)

{

case i:

Print ("Hey enter index:");

scanf ("%d", &var);

Print ("Position of value %d is %d", var, val[var]);

break;

case ii:

Print ("Hey enter element for Position finding");

scanf ("%d", &val);

int result = binary search (val, 0, num-1, var);

(result == -1);

Print ("OPPS sorry not found your element");

Print ("Hurray here is your element at
index %d", result);

return 0;

case iii:

Print ("Enter two indices val");

scanf ("%d %d", &p1, &p2);

Sum = val[p1] + val[p2];

Print ("Sum = %d", sum);

Print (a PRODUCT = $1 \cdot d^u, p_r$);

Srikar

break;

}

}

Rough

In merge sort:-

Input list is

34 99

5 2

57 40

8 29

{34, 99}

{2, 5}

{40, 57}

{8, 29}

{2, 5, 34, 99}

{8, 29, 40, 57}

{2, 5, 8, 29, 34, 40, 57, 99}

arr(s, v)
arr(r+1, i)
merge
(arr(s, r, i))

Sorted list
(In merge sort).

SORTING PROCESS
IN
MERGE SORT.

[Srikanth API allu] 474
Printf ("Please enter your Preference");

scanf ("%d", &OP);
switch (OP)

{

case i;

Printf ("Hey enter index:");

scanf ("%d", &var);

Printf ("Position of value %d is %d", var, arr[var]);

break;

case ii:

Printf ("Hey enter element for Position finding");

scanf ("%d", &val);

int result = binarysearch (val, 0, n-1, arr);

(result == -1);

Printf ("OPPS sorry not found your element");

Printf ("Hurray here is your element at
index %d", result);

return 0;

case iii:

Printf ("Enter two indices val");

scanf ("%d %d", &p1, &p2);

Sum = arr[p1] + arr[p2];

Printf ("Sum = %d", sum);

```

# include <stdio.h>
# include <stdlib.h>

void merge (int ar[], int s, int r, int i)
{
int s, r, i;
int p1 = .
    int K, a, j;
    int p1 = r - s + 1;
    int p2 = i - s;
}

```

/*① Here in this case we should create arrays which are utilized temporality (for problem easy solving).

```
int X [p1], Y [p2];
```

/*② Copy data to above case ↑↑

```
for (k=0; k < p1; k++)
```

```
S[k] = ar [s+i];
```

```
for (a=0; a < p2; a++);
```

```
r[a] = ar [k+1+r]
```

/* Taking K, a = 0 & j = 1

merge temp array and initial index of sub arrays consider.

while ($k < p_1$ || $a < p_2$)

{

if ($s[k] \leq i[a]$)

arr[j] = s[k];

k++;

else

arr[j] = i[k];

a++;

j++;

}

while ($a < p_2$)

{

arr[j] = i[k]

a++;

j++;

}

}

void merge sort (int arr[], int s, int i)

{

if ($s > i$)

{

int x = $s + (i - s) / 2$;

}

}

void print array (int v[], int size)

Snikar

int k;

for (k=0; k<size; k++)

printf ("%d", v[k]);

printf (" ");

}
it
maine)

{

int size, u;

printf ("Enter size (ARRAY): ");

scanf ("%d", &size);

int val[size];

for (v=0; v<size; v++)

{

printf ("Hey please enter Required values:");

scanf ("%d", &val[v]);

}

printf Array (val, size);

merge sort (val, 0, size-1);

printf ("Here is sorted array: ");

printf array (val, size);

~~int j, s,~~

Srikan

int P, q, h, G₁, G₂, temp;

printf("Enter
P
value here:");

scanf("%d", &P);

G₁ = G₂ = 1;

for (q = 0; q <= P; q++)

{ temp = val[q];

G₁ = temp * G₁;

}

for (h = size - 1; h >= P; h--)

{ temp = val[h]

G₂ = temp * G₂;

}

printf("Multiplication of element K
from first and last is required
is:");

printf("%d %d, G₁, G₂);

}

(2)

Srikan

SELECTION SORT:

Principle Identifying Smallest element in list and moving it to starting of the list. This is continued till all elements are sorted in a list.

SMALL EXAMPLE:-

List: \rightarrow

18	3	2	33	21
----	---	---	----	----

Pass SNO.

Comparison

Resultant array

1

18	3	2	33	21
----	---	---	----	----

2	3	18	33	21
---	---	----	----	----

2

2	3	18	33	21
---	---	----	----	----

2	3	18	33	21
---	---	----	----	----

3

2	3	18	33	21
---	---	----	----	----

2	3	18	33	21
---	---	----	----	----

4

2	3	18	33	21
---	---	----	----	----

2	3	18	33	21
---	---	----	----	----

Pass is the smallest element iterating and beginning of list. 4 Passes are needed for 5 elements.

Sort needs.

$$n-1$$

Passes for

n elements.

INSERTION SORT :-

Nika

Principle Sorting of a list by inserting each successive element of elements in previously sorted sub list. Such element needs other elements to be shifted as needed.

Pass

Comparison

Resultant array.

1

18 | 8 | 1 | 11 | 21

3 | 18 | 11 | 21 | 33

2

18 | 3 | 8 | 11 | 21

2 | 3 | 18 | 11 | 21

3

2 | 3 | 18 | 8 | 11 | 21

2 | 3 | 18 | 11 | 21

4

2 | 3 | 18 | 8 | 11 | 21

3 | 2 | 18 | 11 | 21

Current selection

Previously sorted sub array

In above example we need 4 Passes for 5 elements.
So Insertion sort requires $(n-1)$ passes
to sort an array (of n) elements

④

include <stdio.h>

Shikhar

/* In this Program bubble sort is utilized */

void BubbleSort (int ar[], int m)

{
int p, q, temp;

for (p=0; p<m-1; p++)

for (q=0; q<m-1; q++)

if (ar[q] > ar[q+1]) ;

ar[q]

{ temp = ar[q];

ar[q] = ar[q+1];

ar[q+1] = temp;

}

}

int size, s;

printf ("Enter array size:");

printf ("%d", &size);

int ar[size];

for (p=0; p<size; p++)

{
printf ("%d", ar[p]);

printf (" \n");

}

printf (" \n \n MENU \n");

Isdika
Apaloo
Print f(" Divisible number by %d are : \n", n) ;

for (i=0; i<size; i++)

{
 if (arr[i] % n == 0)

 { printf ("%d\t", arr[i]);

 }

}

5

#include <stdio.h>

int binary_search (int arr, int s, int r, int i)

int mid = (s+r)/2

if (s > r)

return -1

if (arr[mid] == i)

return mid;

if (arr[mid] < i)

return binary_search (arr, mid+1, r, i)

else

return binary_search (arr, s, mid-1, i);

}

```
int main (void)
```

```
int a[50], size, pos, val, m;
```

```
printf ("Enter size of array:");
```

```
scanf ("%d", &a[m]);
```

```
printf ("\n enter array elements:");
```

```
for (m = 0; m < size; m++)
```

```
scanf ("%d", &a[m]);
```

```
printf ("Enter search element:");
```

```
scanf ("%d", &val);
```

```
pos = Binary search (a, size, val);
```

```
if (pos < 0)
```

```
printf ("Sorry ops
```

```
not possible (%d, val);
```

```
else
```

```
printf ("Hurray Position is at: %d (%d, val, pos + 1);
```

```
return 0;
```

The End

Page No: 7

Rough

loop {
↓
}

printf

↓

pos = BS (a, 0,
size - 1, val)

↓

if (case)

↓

else (case)

↓

Print statement