

Assignment-6

Sunkavalli Teja

API9110010483

CSE - F

- ① Take the elements from user & sort them in descending order and do the following.
 - ① Using Binary search find the element and the location in array where the element is asked from user
 - ② Ask the user to enter any two locations. Print the sum and product of values at those locations in the sorted array.

Prog

```
#include <stdio.h>
Void Binarysearch();
Void main() → int number[20];
{
    int number[20];
    int i, j, a, n;
    Printf("enter no of elements");
    scanf("%d", &n);
    Printf("enter numbers");
    for (i=0; i<n; i++)
        scanf("%d", &number[i]);
```

```

for (i=0; i<n; i++)
{
    for (j=i+1; j<n; j++)
    {
        if (number[i]<number[j])
        {
            a = number[i];
            number[i] = number[j];
            number[j] = a;
        }
    }
}

```

Printf ("Numbers in descending order");

```

for (i=0; i<n; i++)
{
    printf ("%d \n", number[i]);
}

```

Printf ("enter two locations");

int x, y, sum, product;

scanf ("%d %d", &x &y);

sum = number[x] + number[y];

product = number[x] * number[y];

Printf ("sum of numbers in two locations is %d", sum);

Printf ("product of two numbers is %d", product);

} Binary search();

```
Void Binarysearch()
```

```
{
```

```
int c, first, last, middle, search;
```

```
printf("enter the value to search");
```

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

```
first = 0;
```

```
last = n-1;
```

```
middle = (first + last) / 2;
```

```
while (first <= last) {
```

```
    if (number[middle] < search)
```

```
        first = middle + 1;
```

```
    else if (number[middle] == search)
```

```
    {
```

```
        printf("%d found at %d",  
                search, middle + 1);
```

```
    }
```

```
    else
```

```
        last = middle - 1;
```

```
    middle = (first + last) / 2;
```

```
    if (first > last)
```

```
        printf("%d is not in list", search);
```

```
}
```

- ② sort the array using merge sort where elements are taken from user and find the product of k^{th} elements from first and last where k is taken from user

```
prog #include <stdio.h>

Void mergesort (int a[], inti, intj);
Void merge (int a[], int i1, int i2, int j1, int j2);

int main()
{
    int a[30], n, i;
    printf ("enter no of elements");
    scanf ("%d", &n);
    printf ("enter array elements");
    for (i=0; i<n; i++)
        scanf ("%d", &a[i]);
    mergesort (a, 0, n-1);
    printf ("\n sorted array is:");
    for (i=0; i<n; i++)
        printf ("%d", a[i]);
    int x, Product=1;
    printf (" enter a value");
```

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

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

```
    product = product * a[i];
```

```
printf ("Product of first %d  
elements is %d", x, product);
```

```
for (i = x; i > x; i--)
```

```
    product = product * a[i];
```

```
printf ("Product of last %d  
elements is %d", x, product);
```

```
return 0;
```

```
}
```

```
void mergesort (int a[], int i, int j)
```

```
{
```

```
    int mid;
```

```
    if (i < j)
```

```
    {
```

```
        mid = (i + j) / 2;
```

```
        mergesort (a, i, mid);
```

```
        mergesort (a, mid + 1, j);
```

```
        merge (a, i, mid + 1, j);
```

```
    }
```

```
}
```

```
void merge (int a[], int i, int j, int k,  
            int l)
```

```
{
```

```
    int temp[50];
```

```
    int i, j, k;
```

```

i = i1;
j = j2;
k = 0;
while (i ≤ j1 && j ≤ j2)
{
    if (a[i] < a[j])
        temp[k++] = a[i++];
    else
        temp[k++] = a[j++];
}
while (i ≤ j1)
{
    temp[k++] = a[i++];
}
while (j ≤ j2)
{
    temp[k++] = a[j++];
}
for (i = i1, j = 0; i ≤ j2; i++, j++)
{
    a[i] = temp[j];
}
}

```

- 3) Discuss insertion sort and selection sort with examples.

Insertion sort:

Insertion sort is a sorting mechanism.

where sorted array is built, having one item at a time. The array elements are compared with each other sequentially in some particular order.

ex:

Initial array :

| | | | | |
|----|---|----|----|----|
| 30 | 9 | 16 | 41 | 11 |
|----|---|----|----|----|

| | | | | |
|----|---|----|----|----|
| 30 | 9 | 16 | 41 | 11 |
|----|---|----|----|----|

copy 9

| | | | | |
|----|----|----|----|----|
| 30 | 30 | 16 | 41 | 11 |
|----|----|----|----|----|

shift 30

| | | | | |
|---|----|----|----|----|
| 9 | 30 | 16 | 41 | 11 |
|---|----|----|----|----|

insert 9, copy 16

| | | | | |
|---|----|----|----|----|
| 9 | 30 | 30 | 41 | 11 |
|---|----|----|----|----|

shift 30

| | | | | |
|---|----|----|----|----|
| 9 | 16 | 30 | 41 | 11 |
|---|----|----|----|----|

insert 16, copy 41,
insert 30,

| | | | | |
|---|----|----|----|----|
| 9 | 16 | 30 | 41 | 11 |
|---|----|----|----|----|

copy 11

| | | | | |
|---|----|----|----|----|
| 9 | 16 | 16 | 30 | 41 |
|---|----|----|----|----|

shift 30, 41, 11

| | | | | |
|---|----|----|----|----|
| 9 | 11 | 16 | 30 | 41 |
|---|----|----|----|----|

insert 11.

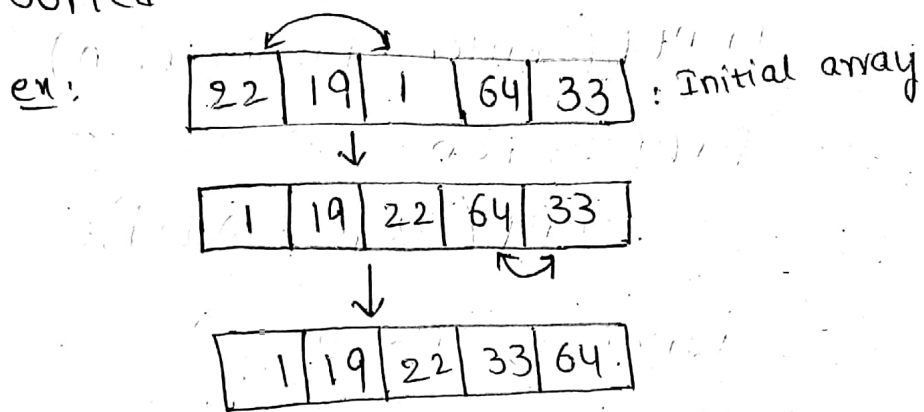
Time complexity:

Best case $\rightarrow O(n)$

Avg, worst case $\rightarrow O(n^2)$.

Selection Sort:

The sorting algorithm, iterates through the array and finds the smallest number in the array and swap it with the first element if it is smaller than the first element. Next, it goes on to second elements and so on until all elements are sorted.



Time Complexity:

Best case $\rightarrow O(n^2)$

Worst case $\rightarrow O(n^2)$

④ Sort the array using bubble sort where elements are taken from user and display elements.

i. In alternate order.

ii. Sum of elements in odd position and

Product of elements in even position.

iii. elements which are divisible by m where m is taken from user.

Prog #include <stdio.h>

int main()

{
int n, temp, i, j, A[30];

printf("enter no. of elements");

scanf("%d", &n);

printf("enter %d number", n);

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

scanf("%d" & A[i]);

for(i=n; i>=0; i--)

{
for(j=0; j<=i; j++)

{

if(A[j] > A[j+1])

{

temp = A[j];

A[j] = A[j+1];

A[j+1] = temp;

}

}

}

printf("sorted elements");

```

for (i=0; i<n; i++)
    printf("%d", A[i]);

int sum=0, product=1;

printf("alternative numbers are\n");
for (i=0; i<n; i++)
{
    if (i%.2 != 0)
        printf("%d\n", A[i]);
}

printf("sum of odd numbers is");
for (i=0; i<n; i++)
{
    if (A[i].2 != 0)
        sum = sum + A[i];
}

printf("%d", sum);

for (i=0; i<n; i++)
{
    if (A[i].2 == 0)
        product *= A[i];
}

printf("Product of even numbers\n");
printf("%d\n", product);

return 0;
}

```

⑤ Write a recursive Program to implement Binary search?

Prog #include <stdio.h>

```
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;
```

```
}
```

```
int main (void)
```

```
{
```

```
    int arr[] = { 2, 3, 4, 10, 40 };
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    int x = 10;
```

```
    int result = Binarysearch (arr, 0, n - 1, x);
```

```
(result == -1)? printf("elements is  
not present in array");
```

```
: printf("elements is present  
at %d", result);
```

```
return 0;
```

```
}
```

Sunkavalli Teja

Ap19110010483

CSE-F.