

DRONACHARYA GROUP OF INSTITUTIONS, GREATER NOIDA

Affiliated to Utter Pradesh Technical University Noida

Approved by AICTE



Algorithms Lab (NCS -551)

INDEX

S.no.	Program
1.	Insertion sort
2.	Merge sort
3.	Bucket sort
4.	Heap sort
5.	Counting sort
6.	Quick sort
7.	Radix sort
8.	Shell Sort
9.	Linear Search
10.	Binary Search

SOLUTIONS

1. PROGRAM TO IMPLEMENT INSERTION SORT.

```
#include<stdio.h>
#include<conio.h>
void main() {
int a[50],i,j,key,n;
clrscr();
printf("\n Enter how many
no:"); scanf("%d",&n);
printf("\n Enter the array
elements:"); for(i=0;i<n ;i++)
scanf("%d",&a[i]);
for(j=1;j<n;j++) {
key=a[j];
i=j-1;
while(i>=0 && a[i]>key)
{
a[i+1]=a[i];
i=i-1;
}
a[i+1]=key;
}
printf("\n Sorted array
is:\n"); for(i=0;i<n;i++)
printf("%d\n",a[i]);
getch();
}
```

OUTPUT-

Enter how many no: 10

Enter the array elements: 18 9 14 13 16 15 19 8 10 4

Sorted array

is: 4 8 9 10 13

14 15 16 18 19

2. PROGRAM TO IMPLEMENT MERGE SORT

```
#include<stdio.h>

#include<conio.h>

#define max 20

voidmergesort(int a[max], intp,int q);

void merge(int a[max],intp,intq,int
r); void main() {

inti,q,r,a[max],n;

clrscr();

printf("Enter how many elements : ");

scanf("%d",&n);

printf("\n Enter the array
elements:"); for(i=1;i<=n;i++)

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

mergesort(a,1,n);

for(i=1;i<=n;i++)

printf("\n%d",a[i]);

getch();

}

voidmergesort(int a[max], int p, int r)

{ int q;

if(p<r) {

q=(p+r)/2;

mergesort(a,p,q);

mergesort(a,q+1,r);

merge(a,p,q,r);

}
```

```

}

void merge(int a[max],int p,int q,int r)

{
int L[max],R[max],i,j,k,n1,n2;

n1=q-p+1;

n2=r-q;

for(i=1;i<=n1;i++)

L[i]=a[p+i-1];

for(j=1;j<=n2;j++)

R[j]=a[q+j];

L[n1+1]=32767;

R[n2+1]=32767;

i=1;

j=1;

for(k=p;k<=r;k++)

{

if(L[i]<R[j])

{

a[k]=L[i];

i=i+1;

}

else

{

a[k]=R[j];

j=j+1;

}

}

}

```

OUTPUT-

Enter how many elements : 10

Enter the array elements:1 9 2 6 3 8 4 5 7 0

0

1

2

3

4

5

6

7

8

9

3. PROGRAM TO IMPLEMENT BUCKET SORT.

```
#include<stdio.h>

#include<conio.h>

void main()

{

int a[10][10],b[10],c[10],d[10];

inti,j,k=0,n,m,s;

clrscr();

printf("\n\n\tEnter the total number of

elements\n\n"); scanf("%d",&n);

printf("\n\nEnter the maximum number of digits in a number\n\n");

scanf("%d",&m);

printf("\n\nEnter

Elements\n\n"); for(i=0;i<n;i++)

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

for(s=1;s<=m;s++)

{

for(i=0;i<10;i++)

{

for(j=0;j<10;j++)

a[i][j]=-1;

}

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

d[i]=b[i];

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

{

k=0;
```



```
While(k!=s) {
c[i]=b[i]%10;
b[i]=b[i]/10;
k++;
} }
for(i=0;i<n;i++) {
j=0;
while(a[c[i]][j]!=-1)
j++;
a[c[i]][j]=d[i];
}
for(i=0;i<n;i++)
d[i]=-1;
k=0;
for(i=0;i<10;i++) {
j=0;
while(a[i][j]!=-1) {
d[k]=a[i][j];
k++;
j++;
}
}
for(i=0;i<n;i++)
b[i]=d[i];
}
printf("\n\nSorted Elements
Are\n\n"); for(i=0;i<n;i++)
printf("\n%d",d[i]);
getch();
}
```

OUTPUT-

Enter the total number of elements

10

Enter the maximum number of digits in a number

2

Enter Elements

78 17 39 26 72 94 21 12 23 68

Sorted Elements Are

12

17

21

23

26

39

68

72

78

94

4. PROGRAM TO IMPLEMENT HEAP SORT.

```
#include <stdio.h>
#define MAX 10
void swap(int *x,int *y) {
int temp;
temp = *x;
*x = *y; *y
= temp;
}
void adjust( int list[],inti, int n)
{ intj,k,flag;
k = list[i];
flag = 1;
j = 2 * i;
while(j <= n && flag) {
if(j < n && list[j] < list[j+1])
j++;
if( k >= list[j])
flag =0;
else {
list[j/2] = list[j];
j = j *2;
} }
list [j/2] = k;
}
voidbuild_initial_heap( int list[], int n)
{
inti;
for(i=(n/2);i>=0;i--)
adjust(list,i,n-1);
}
voidheapsort(int list[],int n)
{
inti;
build_initial_heap(list,n);
for(i=(n-2); i>=0;i--)
{
swap(&list[0],&list[i+1]);
adjust(list,0,i);
}
}
voidreadlist(int list[],int n)
{
inti;
```

```

printf("Enter the elements\n");
for(i=0;i<n;i++)
scanf("%d",&list[i]);
}
void printlist(int list[],int n)
{
inti;
printf("The elements of the list are:
\n"); for(i=0;i<n;i++)
printf("%d\t",list[i]);
}
void main()
{
int list[MAX], n;
clrscr();
printf("Enter the number of elements in the list max=10\n");

scanf("%d",&n);
readlist(list,n);
printf("The list before sorting
is:\n"); printlist(list,n);
heapsort(list,n);
printf("\nThe list after sorting
is:\n"); printlist(list,n);
getch();
}

```

OUTPUT-

Enter the number of elements in the list max = 10

8

Enter the elements

3 4 9 2 1 5 10 0

The list before sorting is:

The elements of the list are:

3 4 9 2 1 5 10 0

The list after sorting is:

The elements of the list are:

0 1 2 3 4 5 9 10

5. PROGRAM TO IMPLEMENT COUNTING SORT.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[20],b[20],c[100],n,i,k,big;
clrscr();
printf("\n how many
number:"); scanf("%d",&n);
printf("\n Enter the array
elements\n"); for(i=1;i<=n;i++)
scanf("%d",&a[i]);
big=a[1];
for(i=2;i<=n;i++)
{
if(big<a[i])
big=a[i];
}
printf("biggest=%d",big);
for(i=0;i<=big;i++)
c[i]=0;
for(k=1;k<=n;k++)
c[a[k]]=c[a[k]]+1;
for(i=1;i<=big;i++)
c[i]=c[i]+c[i-1];
for(k=n;k>=1;k--)
{
b[c[a[k]]]=a[k];
c[a[k]]=c[a[k]]-1;
}
printf("\n sorted
array:"); for(i=1;i<=n;i++)
printf(" %d\t",b[i]);
getch();
}
```

OUTPUT-

How many number:10

Enter the array
elements 8 9 6 5 8 7 6 4
9 3 Biggest=9

Sorted array:
3 4 5 6 6 7 8 8 9 9

6. Wap to implement quick sort.

```
#include<stdio.h>
#include<conio.h>
#define max 100
int a[max],n,i,l,h;
void main() {
    void input(void);
    input();
    getch();
}
void input(void) {
    void quick_sort(int a[],int i,int
h); void output(int a[],int n);
    printf("How many elements in the array :
"); scanf("%d",&n);
    printf("\n");
    printf("Enter the elements :\n");
    for(i=0;i<=n-1;i++) {
        scanf("%d",&a[i]);
    }
    fflush(stdin);
    i=0;
    h=n-1;
    quick_sort(a,i,h);
    printf("sorted array
:\n"); output(a,n);
    fflush(stdout);
}
void quick_sort (int a[],int i,int h) {
    int temp,key,low,high;
    low=i;
    high=h;
    key=a[(low+high)/2];
    do {
        while(key >a[low])
        { low++;
        }
        while(key<a[high]) {
            high--;
        }
        if(low<=high) {
            temp=a[low];
            a[low++]=a[high];
            a[high--]=temp;
        }
    } while(low<=high);
}
```

```
if(l<high)
quick_sort(a,l,high);
if(low<h)
quick_sort(a,low,h);
}
void output(int a[],int n)
{ for(i=0;i<=n-1;i++) {
printf("%d\n",a[i]);
}
}
```

OUTPUT:

How many elements in the array :

4 Enter the elements : 7 41 3 22

Sorted array :

3

7

22

41

7. Wap to implement radix sort.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[5]={343, 562, 785, 287, 321};
int d, temp=0, i=0, j=0, k=0, u[5], t[5],
h[5]; clrscr();
for(i=0; i<5; i++)
{
u[i]=a[i]%10;
d=a[i]/10;
t[j]=d%10;
h[i]=d/10;
}
for(i=0; i<5; i++)
{
for(j=0; j<4; j++)
{
if(u[j+1]<u[j])
{
temp= a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
for(j=0; j<5; j++)
{
for(j=0; j<5; j++)
{
if(u[j+1]<h[j])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
for(i=0; i<5; i++)
{
for(j=0; j<5; j++)
{
if(h[j+1]<h[j])
{
temp= a[j];
```



```
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
for(i=0; i<5; i++)
{
printf("\n %d Sorted array :", a[i]);
}
getch();
}
```

OUTPUT:

Sorted
array: 287
321 343 562
785

8. WAP to implement shell sort

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[25],i,j,k,l,n,temp,z,counter=0;;
clrscr();
printf("Enter the size of the array:");
scanf("%d",&n);
printf("\nEnter the elements:");
for(i=1; i<=n; i++)
{
scanf("%d",&a[i]);
}
for(i=3; i>=1; i--)
{
for(j=1; j<=i; j++)
{
for(k=j; k<=n; k=k+i)
{
for(l=k+i; l<=n; l=l+i)
{
if(a[k]>a[l])
{
temp=a[k];
a[k]=a[l];
a[l]=temp;
}
}
}
}
}
}
```