­­­­­­­­­ P. Jaswanth Roy

AP19110010524

CSE-H

DSA Lab Programs

# 1. Write a c program for insertion sort algorithm.

# Program:

#include<stdio.h>

int main()

{

int i, j, count, temp, number[25];

printf("enter number of elements in the array: ");

scanf("%d",&count);

printf("Enter %d elements: ", count);

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

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

for(i=1;i<count;i++){

temp=number[i];

j=i-1;

while((temp<number[j])&&(j>=0)){

number[j+1]=number[j];

j=j-1;

}

number[j+1]=temp;

}

printf("Order of Sorted elements: ");

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

printf(" %d",number[i]);

return 0;

}

# Output:

Enter number of elements in the array: 5

Enter 5 elements:5

3

4

2

1

Order of sorted elements: 123

# 2. Write a c program for selection sort algorithm.

# Program:

#include <stdio.h>

int main()

{

int array[100], n, i, d, position, t;

printf("Enter number of elements:");

scanf("%d", &n);

printf("Enter %d integers:",n);

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

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

for (i = 0; i < (n - 1); i++)

{

position = i;

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

{

if (array[position] > array[d])

position = d;

}

if (position != i)

{

t = array[i];

array[i] = array[position];

array[position] = t;

}

}

printf("Sorted list in ascending order:\n");

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

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

return 0;

}

# Output:

enter the number of elements : 5

enter 5 integers: 60

45

30

78

90

Sorted list in ascending order:

30

45

60

78

90

# 3. Write a c program for bubble sort algorithm.

# Program:

#include <stdio.h>

int main()

{

int array[100], n, i, d, swap;

printf("Enter number of elements:");

scanf("%d", &n);

printf("Enter %d integers:", n);

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

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

for (i = 0 ; i < n - 1; i++)

{

for (d = 0 ; d < n - i - 1; d++)

{

if (array[d] > array[d+1])

{

swap = array[d];

array[d] = array[d+1];

array[d+1] = swap;

}

}

}

printf("Sorted list in ascending order:\n");

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

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

return 0;

}

## Output:

Enter the number of elements: 5

Enter 5 integers: 11

54

99

45

60

Sorted list in ascending order:

11

45

54

60

99

# 4. Write a c program for merge sort algorithm

# Program:

#include <stdio.h>

void merge\_sort(int i, int j, int a[], int aux[]) {

if (j <= i) {

return;

}

int mid = (i + j) / 2;

merge\_sort(i, mid, a, aux);

merge\_sort(mid + 1, j, a, aux);

int pointer\_left = i;

int pointer\_right = mid + 1;

int k;

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

if (pointer\_left == mid + 1)

{

aux[k] = a[pointer\_right];

pointer\_right++;

}

else if (pointer\_right == j + 1)

{

aux[k] = a[pointer\_left];

pointer\_left++;

}

else if (a[pointer\_left] < a[pointer\_right])

{

aux[k] = a[pointer\_left];

pointer\_left++;

}

else

{

aux[k] = a[pointer\_right];

pointer\_right++;

}

}

for (k = i; k <= j; k++)

{

a[k] = aux[k];

}

}

int main()

{

int a[100], aux[100], n, i, d, swap;

printf("Enter number of elements in the array:");

scanf("%d", &n);

printf("Enter %d integers:", n);

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

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

merge\_sort(0, n - 1, a, aux);

printf("Printing the sorted array:\n");

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

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

return 0;

}

# Output:

Enter the number of elements in the array: 7

Enter 7 integers: 45

78

62

12

99

9

10

Printing the sorted array:

9

10

12

45

62

78

99

# 5. Write a c program for Heap Sort algorithm

# Program:

#include<stdio.h>

void create(int []);

void down\_adjust(int [],int);

int main()

{

int heap[30],n,i,last,temp;

printf("Enter no. of elements:");

scanf("%d",&n);

printf("\nEnter %d elements:",n);

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

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

heap[0]=n;

create(heap);

while(heap[0] > 1)

{

last=heap[0];

temp=heap[1];

heap[1]=heap[last];

heap[last]=temp;

heap[0]--;

down\_adjust(heap,1);

}

printf("\nArray after sorting:\n",n);

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

printf("%d ",heap[i]);

return 0;

}

void create(int heap[])

{

int i,n;

n=heap[0];

for(i=n/2;i>=1;i--)

down\_adjust(heap,i);

}

void down\_adjust(int heap[],int i)

{

int j,temp,n,flag=1;

n=heap[0];

while(2\*i<=n && flag==1)

{

j=2\*i;

if(j+1<=n && heap[j+1] > heap[j])

j=j+1;

if(heap[i] > heap[j])

flag=0;

else

{

temp=heap[i];

heap[i]=heap[j];

heap[j]=temp;

i=j;

}

}

}

# Output:

Enter no. of elements: 5

Enter 5 elements: 9

6

4

8

1

Array after sorting:

1

4

6

8

9