Md. Asad Chowdhury Dipu

ID: 2019-1-60-093

Section: 01

Quick Sort

#include<stdio.h>

void swap(int\* a, int\* b)

{

int t = \*a;

\*a = \*b;

\*b = t;

}

//=start, E= End, P= Pivot

//Algorithm 2 Lomuti Partitioninhg Scheme

int partition (int arr[], int S, int E)

{

int P = arr[E];

int i = (S - 1);

for (int j = S; j <= E- 1; j++)

{

if (arr[j] <= P)

{

i++;

swap(&arr[i], &arr[j]);

}

}

swap(&arr[i + 1], &arr[E]);

return (i + 1);

}

void QSort(int arr[], int S, int E)

{

if (S < E)

{

int pi = partition(arr, S, E);

QSort(arr, S, pi - 1);

QSort(arr, pi + 1, E);

}

}

void printArray(int arr[], int size)

{

int i;

}

int main()

{

int n, arr[50];

printf("Enter the amount of elements N: ");

scanf("%d",&n);

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

{

printf("\nEnter Item %d :",i+1);

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

}

QSort(arr, 0, n-1);

printf("The sorted array is: ");

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

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

printf("\n");

return 0;

}

# Merge Sort

#include<stdio.h>

void Merge(int arr[],int mid, int S,int End)

{

int i,j,C,Arr2[20];

i=S;

j=mid+1;

C=S;

while(i<=mid && j<=End)

{

if(arr[i]<arr[j])

{

Arr2[C]=arr[i];

i++;

C++;

}

else

{

Arr2[C]=arr[j];

j++;

C++;

}

}

while(i<=mid)

{

Arr2[C]=arr[i];

C++;

i++;

}

while(j<=End)

{

Arr2[C]=arr[j];

C++;

j++;

}

for(int x=S; x<=End; x++)

{

arr[x]=Arr2[x];

}

}

void MSort(int arr[],int S,int End)

{

int mid;

if(S<End)

{

mid=((S+End)/2);

MSort(arr,S,mid);

MSort(arr,mid+1,End);

Merge(arr,mid,S,End);

}

}

int main()

{

int arr[50],n;

printf("Enter the amount of elements N: ");

scanf("%d",&n);

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

{

printf("\nEnter Item %d :",i+1);

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

}

MSort(arr,0,n-1);

printf("The sorted array is: ");

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

{

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

}

return 0;

}