Wrashall's

shortest path

for(int k=0;k<v;k++){

for(int i=0;i<v;i++){

for(itn j=0;j<v;j++){

if(g[i][k]+g[k][j]<g[i][j]){

g[i][j]=g[i][k]+g[k][j];

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

binary search:-

binary(int []array,int x,int l,int h){

while(l<h){

int mid=l+(h-l)/2;

if(array[mid]==x)

return mid;

else if(array[mid]<x)

l=mid+1;

else

h=mid-1;

}

return -1;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

insertion sort:-

insertion(int []ar){

for(int i=1;i<size;i++){

int idx=ar[i];

int j=i-1;

while(j>=0 && idx<ar[j]){

ar[j+1]=a[j];

--j;

}

ar[j+1]=idx;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

selection sort:-

selection(int []ar){

for(int i=0;i<size-1;i++){

int idx=i;

for(int j=i+1;j<size;j++){

if(ar[j]<ar[idx]){

idx=j;}

}

swap(ar[i],ar[idx]);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

quick sort:-

quick(int[]ar,int l,int h){

if(l<h){

int pi=sort(ar,l,h);

quick(ar,l,pi-1);

quick(ar,pi+1,h);

}

sor t(int []ar,int l,int h){

int p=ar[h];

int i=(l-1);

for(int j=low;j<=h;j++){

if(arr[j]<=p){

i++;

swap(ar[i],ar[j]);}}

swap(ar[i+1],ar[h]);

return i+1;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dutch National Flag:-

void DNF(int []ar,int size){

int low=0,mid=0,high=size;

while(mid<=high){

switch(ar[mid]){

case 0: swap(ar[low],ar[mid]);

low++;mid++;break;

case 1: mid++;break;

case 2: swap(ar[mid],ar[high]);

high--;break;}}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Merge Sort:-

search(int []ar,int l,int m,int r){

int n1=l-m+1;

int n2=r-l;

int L[]=new int[n1];

int R[]=new int[n2];

for(int i=0;i<n1;i++)

L[i]=ar[l+i];

for(int i=0;i<n2;i++)

R[i]=ar[m+1+j];

int i=0,j=0,k=l;

while(i<n1&&j<n2){

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

arr[k]=L[i];

i++;}

else{

arr[k]=M[j];

j++;}k++;}

while(i<n1){

arr[k]=L[i];i++;k++;}

while(j<n2){

arr[k]=L[j];j++;k++;}}}