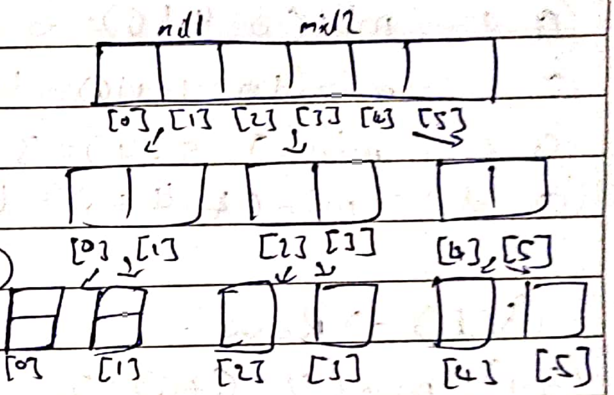
```
int main()
{
 int data[]={45,-2,-45,78,30,-42,10,19,73,93}, i;
 mergesort3way(data, 10);
  printf("After 3way mergesort \n");
    for( i=0; i<10; i++)
         printf(" %d \n", data[i]);
  return 0;
}
```



```
void mergesort3way(int garr[], int n)
{  int i
     if (n==0)
        return;
 int farr[n];
  for(i=0;i<n;i++)          // duplicate array
       farr[i]= garr[i];
 mergesort3wayrec(farr, 0, n, garr);

  for(i=0;i<n;i++)                    // Copy back elements of
        garr[i]= farr[i];             // duplicate array to original array
}

void mergesort3wayrec(int garr[], int low, int high, int destarr[])
{
    if(high-low<2) // size of array atleast(3)
      return;
```

Teacher's Signature..............................

```
int mid1 = low +((high-low)/3);  // arr[1] in first recurssion
int mid2= low+2 *((high-low)/3)+1;//arr[3] in first recurssion


mergesort3wayrec(destarr, low, mid1, garr);
mergesort3wayrec(destarr, mid1, mid2, garr);
mergesort3wayrec(destarr, mid2, high, garr);

merge(destarr, low, mid1, mid2, high, garr);
}

void merge(int garr[], int low, int mid1, int mid2, int high,
            int destarr[])
{
   int i=low, j=mid1, k=mid2, l=low;

while((i<mid1) && (j<mid2) && (k<high))
{
      if (garr[i]<gar[j]
       {
         if (gar[i]<gar[k]
         {
            destarr[l++]= gar[i++];
          }
         else
         {
            destarr[l++]=garr[k++];
         }
      }
     else
     {
      } destarr[l++]= garr[k++];
} }
```

```
while((i<mid1) && (j<mid2))  // First and second

 if (garr[i]< garr[j])
  {
     destarr[l++] = garr[i++];
   }
  else
  {
    destarr[l++]= garr[j++];
  }

while ((j<mid2) && (k<high))  // second and ~~first~~ third
{
     if (garr[j] < garr[k])
      {
          dest arr[l++] =garr[j++];
      }
     else
      {
          destarr[l++] = garr[k++];
      }
 }
while((i<mid1) && (k<high))  // First and third
 {
    if(garr[i] < garr[k])
    {
      destarr[l++] = garr[i++];
     }
   else
   {
     dest arr[l++] =garr[k++];
   }
}

while (i<mid1)
     destarr[l++]= garr[i++];
while (j < mid2)
     destarr[l++]= garr[j++];
while (k<high)
     destarr[l++] = garr[k++];
}
```

CREDIT

GeeksforGeeks