|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | Compiled date | Submitted date | |  |  |  | |

*Quick Sort Algorithm*

*QuickSort(A,p,r)*

1. If p<r
2. then q<-PARTITION(A,p,r)
3. QUICKSORT(A,p,q-1)
4. QUICKSORT(A,q+*1,r)*

*Partitioning*

Partition subarray A[p…….r]by the following procedure:

*PARTITION(A,p,r)*

1. X<-A[r]
2. I<-p-1
3. For i<-p to r-1
4. If A[j]<=x
5. I <- i+1
6. Exchange A[i]<->A[j]
7. exchange A[i+1] <-> A[r]
8. return i+1

*QUICKSORT IMPLEMENTATION*

|  |
| --- |
| #include<stdio.h> |
|  | #include<conio.h>  #include<process.h>  #include<alloc.h> |
|  | int count=0; |
|  | int a; |
|  | int main() |
|  | { |
|  | void quick\_sort(int a[],int, int); |
|  | void getdata(int a[],int n); |
|  | void putdata(int a[],int n); |
|  | int a[100],n; |
|  | printf("enter the size of an array:\n"); |
|  | scanf("%d",&n); |
|  | printf("enter the elements of array:\n"); |
|  | getdata(a,n); |
|  | quick\_sort(a,1,n); |
|  | putdata(a,n); |
|  | printf("\n for n =%d value od count is %d",n,count); |
|  | return 0; |
|  | } |
|  | int partition(int a[],int p,int r) |
|  | { |
|  | int pivot,i,j; |
|  | pivot=a[r]; |
|  | i= p-1; |
|  | count++; |
|  | for(j=p;j<=r-1;j++) |
|  | { |
|  | count++; |
|  | if(a[j]<=pivot) |
|  | { |
|  | count++; |
|  | i=i+1; |
|  | count++; |
|  | temp=a[i]; |
|  | a[i]=a[j]; |
|  | a[j]=temp; |
|  | count++; |
|  | } |
|  | } |
|  | temp=a[i+1]; |
|  | a[i+1] = a[r]; |
|  | a[r]=temp; |
|  | return i+1; |
|  | } |
|  | void quick\_sort(int a[],int p,int r) |
|  | { |
|  | int q; |
|  | if(p < r) |
|  | { |
|  | count++; |
|  | q = partition(a,p,r); |
|  | count++; |
|  | quick\_sort(a,p,q-1); |
|  | count++; |
|  |  |
|  | quick\_sort(a,q+1,r); |
|  | count++; |
|  |  |
|  | } |
|  | } |
|  | void getdata(int a[],int n) |
|  | { |
|  | int i; |
|  | for(i=1;i<=n;i++) |
|  | { |
|  | scanf("%d", &a[i]); |
|  |  |
|  | } |
|  | } |
|  | void putdata(int a[],int n) |
|  | { |
|  | int i; |
|  | for(i = 1;i<=n;i++) |
|  | printf("%d",a[i]); |
|  | printf("\n"); |
|  | } |

*Graph of QUICKSORT*

|  |  |  |  |
| --- | --- | --- | --- |
| Input Size | Best Case | Average Case | Worst Case |
| 5 | 60 | 45 | 42 |
| 10 | 225 | 129 | 150 |
| 15 | 490 | 135 | 322 |
| 20 | 855 | 237 | 555 |
| 25 | 1420 | 274 | 852 |