实验七 排序

一、实验环境：

Windows 10 64位 MinGW GCC 4.8.1

IDEA：CLoin

二、实验内容：

编写程序实现快速排序。

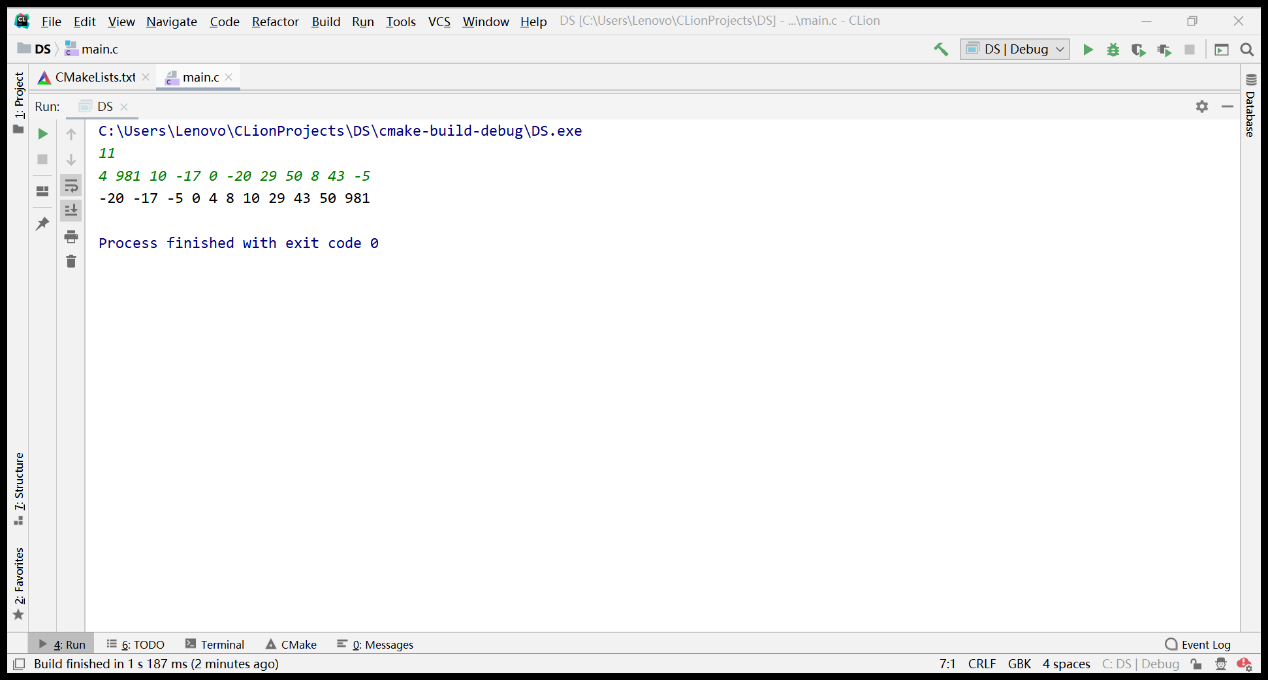
三、算法流程：

输入一段序列对其进行快速排序。

四、程序代码：

#include **<stdio.h>  
void** print(**int** a[],**int** n){  
 printf(**"%d"**,a[1]);  
 **for**(**int** i=2;i<=n;i++){  
 printf(**" %d"**,a[i]);  
 }  
 printf(**"\n"**);  
}  
**int** ask(**int** a[],**int** l,**int** mid,**int** r){  
 **if**(a[l]>a[mid]){  
 mid^=l;  
 l^=mid;  
 mid^=l;  
 }  
 **if**(a[mid]>a[r]){  
 mid^=r;  
 r^=mid;  
 mid^=r;  
 }  
 **if**(a[l]>a[mid]){  
 mid^=l;  
 l^=mid;  
 mid^=l;  
 }  
 **return** mid;  
}  
**void** swap(**int** \*a,**int** \*b){  
 **int** t=\*a;  
 \*a=\*b;  
 \*b=t;  
}  
**void** Qsort(**int** a[],**int** l,**int** r,**int** n ){  
 **if**(l>=r)  
 **return**;  
 **int** mid=(l+r)>>1;  
 **int** p=ask(a,l,mid,r);  
 swap(&a[p],&a[r]);  
 **int** L=l;  
 **int** R=r-1;  
 **while**(L<R){  
 **while**(a[L]<a[r])  
 L++;  
 **while**(a[R]>a[r])  
 R--;  
 **if**(L<R)  
 swap(&a[L],&a[R]);  
 }  
 swap(&a[L],&a[r]);  
*// printf("%d %d\n",l,r);  
// print(a,n);* Qsort(a,l,L,n);  
 Qsort(a,L+1,r,n);  
}  
**int** main() {  
 **int** n;  
 scanf(**"%d"**,&n);  
 **int** a[n+1];  
 **for** (**int** i=1;i<=n;++i) {  
 scanf(**"%d"**,&a[i]);  
 }  
 Qsort(a,1,n,n);  
 print(a,n);   
 **return** 0;  
}

五、运行结果截图：



六、注意事项：

快速排序实质是基于一种分类的思想，如何选取枢纽轴会极大的影响到排序的效率。显然当选取到的枢纽轴是区间的中位数时效率最高。