

## 思路

- 找一个基准（比如第一个元素或者随机元素），将元素分成左右两半，左边小于等于基准，右边大于等于基准；
- 对左右两半分别快速排序；
- 排序结束，元素已经有序了。

这其中，比较困难的是如何将元素分为两半。假设有一组数：3 4 2 5 1，我们

- 取基准元素base=3，设置左边指针i指向起始数3，右边指针j指向最后的数1，然后把这个数（即3）挖掉空着，即[] 4 2 5 1；
- 随后，从j往前找第一个比base小的数（即1）填到[]上，然后这个数也被挖掉，变成1 4 2 5 []，j自减指向前一个数5；
- 然后再从i向右找第一个比base大的数（即4）填到[]上，然后这个数也被挖掉，则有1 [] 2 5 4，i指向后一个数2；
- 然后重复上述步骤，变成1 2 [] 5 4；
- 这时i和j指向同一个位置[]，令其为base，即变成1 2 3 5 4；
- 最后对左右两边的子数组分别排序。

## 代码

```
#include<iostream>
using namespace std;

void quick_sort(int* num, int left, int right){

    if(left >= right)
        return;

    int i = left;
    int j = right;
    int base = num[left];

    while(i < j){

        while(i < j && num[j] >= base)j--;
        if(i < j)
            num[i++] = num[j];

        while(i < j && num[i] < base)i++;
        if(i < j)
            num[j--] = num[i];
    }

    num[i] = base;
    quick_sort(num, left, i-1);
    quick_sort(num, i+1, right);
}
```

```
int main(){

    int n;

    while(cin >> n && n != 0){
        int num[n];
        for(int i = 0; i < n; i++)
            cin >> num[i];

        quick_sort(num, 0, n-1);

        for(int i = 0; i < n; i++)
            cout << num[i] << " ";
        cout << endl;
    }

    return 0;
}
```