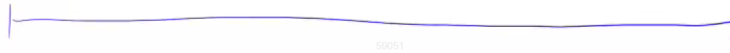


快排



① 确定分界点: $q[l]$, $q[(l+r)/2]$, $q[r]$ 随机

② 调整区间: $\leq x$ $\geq x$ ☆

③ 递归处理左右两段



```
1  #include <iostream>
2
3  using namespace std;
4
5  const int N = 1e6 + 10;
6
7  int n;
8  int q[N];
9
10 void quick_sort(int q[], int l, int r)
11 {
12     if (l >= r) return;
13
14     int x = q[l], i = l - 1, j = r + 1;
15     while (i < j)
16     {
17         do i ++ ; while (q[i] < x);
18         do j -- ; while (q[j] > x);
19         if (i < j) swap(q[i], q[j]);
20     }
21
22     quick_sort(q, l, j);
23     quick_sort(q, j + 1, r);
24 }
25
26 int main()
27 {
28     scanf("%d", &n);
29     for (int i = 0; i < n; i ++ ) scanf("%d", &q[i]);
30
31     quick_sort(q, 0, n - 1);
32
33     for (int i = 0; i < n; i ++ ) printf("%d ", q[i]);
34
35     return 0;
36 }
```

#include

using namespace std;

const int N = 100010;

int q[N];

void quick_sort(int q[], int l, int r)

{

if (l >= r) return;

```
int i = l - 1, j = r + 1, x = q[l + r >> 1];
while (i < j)
{
    do i ++ ; while (q[i] < x);
    do j -- ; while (q[j] > x);
    if (i < j) swap(q[i], q[j]);
}

quick_sort(q, l, j);
quick_sort(q, j + 1, r);
```

}

int main()

{

int n;

scanf("%d", &n);

```
for (int i = 0; i < n; i ++ ) scanf("%d", &q[i]);
```

```
quick_sort(q, 0, n - 1);
```

```
for (int i = 0; i < n; i ++ ) printf("%d ", q[i]);
```

```
return 0;
```

}

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/39784/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。