

排序问题(1s,512MiB, mphone)

【算法分析】

以 W_i 为第一关键字, 编号为第二关键字排序, 排序完后在对一到十类的人加上 E_i 后在进行一次排序, 规则和第一次排序相同, 最后输出 W_i 从高到低的人的编号。

【核心代码】

```
struct node {
    int num, w;
};
//以权值为第一关键字排序, 编号为第二关键字排序
bool cmp(node x, node y) {
    return x.w > y.w || (x.w == y.w && x.num < y.num);
}
node p[N];
int E[15], n, k;
int main() {
    freopen("mphone.in", "r", stdin);
    freopen("mphone.out", "w", stdout);
    n = read(), k = read();
    for(int i = 1; i <= 10; i++)
        E[i] = read();
    for(int i = 1; i <= n; i++)
        p[i].w = read(), p[i].num = i;
    // 快速排序
    sort(p + 1, p + 1 + n, cmp);
    for(int i = 1; i <= n; i++)
        p[i].w += E[(i - 1) % 10 + 1];
    sort(p + 1, p + 1 + n, cmp);
    //输出前 k 个人的编号
    for(int i = 1; i < k; i++)
        printf("%d ", p[i].num);
    printf("%d", p[k].num);
    return 0;
}
```

【参考程序】

```
#include<cstdio>
#include<algorithm>
using namespace std;
const int N = 5e4 + 500;
typedef long long ll;

inline int read() {
    int x = 0, f = 1;
    char c = getchar();
    while(c > '9' || c < '0') {
```

```

        if(c == '-')    f = -f;
        c = getchar();
    }
    while(c >= '0' && c <= '9') {
        x = x * 10 + c - '0';
        c = getchar();
    }
    return x * f;
}

struct node {
    int num, w;
};
//以权值为第一关键字排序，编号为第二关键字排序
bool cmp(node x, node y) {
    return x.w > y.w || (x.w == y.w && x.num < y.num);
}
node p[N];
int E[15], n, k;

int main() {
    freopen("mphone.in", "r", stdin);
    freopen("mphone.out", "w", stdout);
    n = read(), k = read();
    for(int i = 1; i <= 10; i++)
        E[i] = read();
    for(int i = 1; i <= n; i++)
        p[i].w = read(), p[i].num = i;
    // 快速排序
    sort(p + 1, p + 1 + n, cmp);
    for(int i = 1; i <= n; i++)
        p[i].w += E[(i - 1) % 10 + 1];
    sort(p + 1, p + 1 + n, cmp);
    //输出前 k 个人的编号
    for(int i = 1; i <= k; i++)
        printf("%d ", p[i].num);
    return 0;
}

```