贪心

贪心算法是指在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，只做出在某种意义上的局部最优解。贪心算法不是对所有问题都能得到整体最优解，关键是贪心策略的选择，选择的贪心策略必须具备无后效性，即某个状态以前的过程不会影响以后的状态，只与当前状态有关。

解题的一般步骤是

1. 数学模型来描述问题
2. 把求解的问题分成若干个子问题
3. 对每一子问题求解，得到子问题的局部最优解
4. 把子问题的局部最优解合成原来问题的一个解。

练习题

P1223 排队接水

解题思路

先确定怎么排队接水用时最少，都能想到 ti 小的总用时最少。

证明：

设 a 为第一个人接水时间，b 为第二个人接水时间，且`a < b`则有

`t1 = a + a + b`、`t2 = b + b + a`，显然`t1 < t2`，结论成立

代码

```cpp
#include <iostream>
#include <cstdio>
#include <algorithm>

using namespace std;

// ind 存编号，ti 存接水时间
struct Node {
    int ind;
    int ti;
} arr[1005];
```

```cpp

bool cmp(Node a, Node b) {
    if (a.ti == b.ti) return a.ind < b.ind;
    return a.ti < b.ti;
}

int main() {
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> arr[i].ti;
        arr[i].ind = i + 1;
    }
    sort(arr, arr + n, cmp);
    int temp = 0;
    double sum = 0;
    for (int i = 0; i < n; i++) {
        sum = sum + temp;
        temp += arr[i].ti;
        if (i != 0) cout << " ";
        cout << arr[i].ind;
    }
    printf("\n%.2lf\n", sum / (n * 1.0));
    return 0;
}
```

P1094 纪念品分组

解题思路

尽量把最小的和最大的放在一组，如果最大的过大就自己一组

代码

```cpp
#include <iostream>
#include <cstdio>
#include <algorithm>

using namespace std;

int main() {
    int ans = 0, n, max, num[30005] = {0};
    scanf("%d %d", &max, &n);
```

```
    for (int i = 0; i < n; i++) {
        scanf("%d", &num[i]);
    }
    sort(num, num + n);
    int l = 0, r = n - 1;
    while (l <= r) {
        if (num[l] + num[r] <= max) l++;
        r--;
        ans++;
    }
    printf("%d\n", ans);
    return 0;
}
```

P1031 均分纸牌

解题思路

1. 算出平均值（每堆应为多少张）
2. 当前堆的纸牌数不等于平均值时，从右侧堆取放

代码

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

int main() {
    int arr[105] = {0}, n, x, ans = 0;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
        x += arr[i];
    }
    x /= n;
    // 应为总数为 N 的倍数，最后一堆只能从左侧堆操作，所以不用遍历到最后一堆
    for (int i = 0; i < n - 1; i++) {
        if (arr[i] - x)  {
            arr[i + 1] += (arr[i] - x);
            ans += 1;
        }
    }
    cout << ans << endl;
```

```
20      return 0;
21  }
```

P1080 国王游戏

所用知识

贪心、高精度（除法、乘法）

解题思路

第一种情况

name	l	r
k	a0	b0
p1	a1	b1
p2	a2	b2

可得获奖最多的大臣为：`ans1 = max(a0 / b1, a0 * a1 / b2)`

第二种情况

name	l	r
k	a0	b0
p2	a2	b2
p1	a1	b1

可得获奖最多的大臣为：`ans2 = max(a0 / b2, a0 * a2 / b1)`

进行参数替换后：

- `ans1 = max(k1, k2)`
- `ans2 = max(k3, k4)`

可以看出 `k2 > k3`、`k4 > k1`，如果`ans1 < ans2`，那么`k4 > k2`，即`a1 * b1 < a2 * b2`

则可以根据`ai * bi`的大小进行排序，将`ai * bi`的值小的放在前面，ans 的值才会小

代码

```cpp
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>

using namespace std;

const int MAX_SIZE = 10005;

struct Node {
    int l, r, x;
} num[MAX_SIZE];

bool cmp(Node a, Node b) {
    return a.x < b.x;
}

// 初始化 tl, ans 数组
void init(int *tl, int *ans, int key) {
    tl[0] = ans[0] = 1;
    tl[1] = key;
    return ;
}

// 大数乘法
void mul(int *arr, int x) {
    for (int i = 1; i <= arr[0]; i++) arr[i] *= x;
    for (int i = 1; i <= arr[0]; i++) {
        if (arr[i] < 10) continue;
        arr[i + 1] += arr[i] / 10;
        arr[i] %= 10;
        arr[0] += (i == arr[0]);
    }
    return ;
}

// 大数除法
void div(int *ret, int *arr, int x) {
    int temp = 0;
    for(int i = arr[0]; i >= 1; i--) {
        temp = temp * 10 + arr[i];
        ret[i] = temp / x;
        if (ret[0] == 0 && ret[i] != 0) ret[0] = i;
```

```
        temp %= x;
    }
    return ;
}

void copy(int *a, int *b) {
    memset(b, 0, sizeof(b));
    b[0] = a[0];
    for (int i = a[0]; i >= 1; i--) b[i] = a[i];
    return ;
}

// 比较 tl 和 ans 的大小
void mymax(int *tl, int *ans, int x) {
    int temp[MAX_SIZE] = {0};
    div(temp, tl, x);
    if (temp[0] > ans[0]) copy(temp, ans);
    if (temp[0] == ans[0]) {
        for (int i = temp[0]; i >= 1; i--) {
            if (ans[i] >= temp[i]) continue;
            copy(temp, ans);
            break;
        }
    }
    return ;
}

void output(int *arr) {
    for (int i = arr[0]; i > 0; i--) {
        cout << arr[i];
    }
    cout << endl;
    return ;
}

int main() {
    int n, gl, gr;
    int tl[MAX_SIZE] = {0}, ans[MAX_SIZE] = {0};
    cin >> n >> gl >> gr;
    for (int i = 0; i < n; i++) {
        cin >> num[i].l >> num[i].r;
        num[i].x = num[i].l * num[i].r;
    }
    init(tl, ans, gl);

```

```
89      sort(num, num + n, cmp);
90      for (int i = 0; i < n; i++) {
91          mymax(tl, ans, num[i].r);
92          mul(tl, num[i].l);
93      }
94      output(ans);
95      return 0;
96  }
```