



中国科学院大学
University of Chinese Academy of Sciences

算法设计作业答疑

贪心策略部分

授课老师 卜东波

问题描述：一条直线上有N只猴子，N根香蕉。猴子每一步可以向左或向右移动一步。每只猴子只拿一根香蕉，每根香蕉只能给一只猴子。求猴子与香蕉的匹配，使得尽快使得所有猴子都有香蕉。

输入：两个整数数组，长度均为N。代表猴子与香蕉的位置。

输出：整数，最小时间。

观察：从两只猴子开始。猴子的路径最好不要有交叉。

算法：将猴子和香蕉分别按位置排序。第一个猴子拿第一根香蕉，第二个猴子拿第二根香蕉，依此类推。

复杂度： $O(n\log n)$ ，n为猴子的个数。

正确性：若不是按照贪心规则分配香蕉，则定会有两只猴子的路径交叉。可以证明这两只猴子交换香蕉后，两只猴子的用时一定小于等于之前的用时。

问题描述：决定n个任务J1,J2,...,Jn的顺序。一台服务器和n台PC上能最快的处理完这些任务。任务首先由服务器预处理，之后由PC处理。每个任务互不相关。一台服务器或一台PC同时只能处理一个任务。

输入：两个长度为n的浮点数数列，p和f。分别为n个任务在服务器和PC上需要的处理时间。

输出：包含J1,J2,...,Jn的一个数列

观察：总体时间的组成。服务器的工作时间（恒定的）+ 服务器工作完毕后PC所用的时间。若在PC上需要的时间越短，越排在后面。

注意：总时间不是 $\left(\sum_{i=1}^n p_i\right) + \min(f_i)$

复杂度：nlogn

正确性：若最后一个任务A的PC处理时间不为最短，将其与之前的一个PC处理时间更短的B排在A后。

问题描述：一群人乘船渡河，单个人的体重不会超过船的载重，每条船最多载两人并且不能超出船的载重，最少使用多少条船才能让所有人过河。

输入：人数 n ，船的载重 l ，以及人的体重 $p[i]$, $i=1, 2, \dots, n$ 。

输出：需要船的数目

观察：让体重大的人尽量和体重小的人凑成对坐船

算法：将人按照体重排序，两指针分别从头和尾遍历直到相遇，两个人能一起坐船，指针同时移动，两人坐不下，体重大的人单独坐船，指针移动一位。

注意：还有另外一种更直观的贪心做法，但复杂度高。

复杂度： $O(n\log n)$ ， n 为人数，排序复杂度。

正确性：

1. 先按照直观的贪心做法来做：将人按体重从大到小排列后 p_1, p_2, \dots, p_n ，若第一个人的体重足够重，只能单独做船。则 $\text{Opt}(p_1, p_2, \dots, p_n) = \text{Opt}(p_2, \dots, p_n) + 1$ 。
2. 否则，从剩下的人重挑出最重的那个人 p_m 与第一个人共坐一条船，此时 $\text{Opt}(p_1, p_2, \dots, p_n) = \text{Opt}(p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_n) + 1$ 。
3. 然后，证明标准贪心做法可以转换成此贪心做法。若有两条小船，分别为 (p_1, p_2) , (p_3, p_4) ，其中 $p_1 > p_2 > p_3 > p_4$ ，则可证明可以交换 p_2 与 p_4 ，成为 (p_1, p_4) , (p_3, p_2) 。

问题描述：给定1到n的一个排列，求如何将其分割成k份后，使得每个分割里的最大值的和最大。

输入：1-n的一个排列，分割数k

输出：使得和最大的k分割，以及这样的分割的个数

观察：值最大的数单独分割

算法：将所有数排序后，得到其最大的k个值，使其处于独立的k个分割中，剩下的数随意分配。具体的分割方法是每个剩余的数的分配方法的乘积。

注意：还有另外一种更直观的贪心做法，但复杂度高。

复杂度： $O(n \log n)$ ，n为数的个数。

正确性：若不按照所述方法划分k分割，不妨设第i大的值为一个区域的最大值，且和第j大的值在一个区域 ($i < j \leq k$)，则可将这个区域中第i大的值两边的值分给其两边的区域（若这个区域在分割的一端，则分一边的区域即可），这个区域的最大值不会变，其他区域的最大值至少不会变小。

问题描述： N 个男孩，第 i 个男孩所在队伍至少有 a_i 个人，求最多可组成的队伍数。

输入： 整数数组 a ，长度为 N 。

输出： 整数，最多队伍数。

观察： a_i 小的男孩互相组队，一个队伍中最大的 a_i 等于队伍人数。

算法： 将数组 a 排序，依次判断第 i 个数对应的男孩是否可作为队伍尾。

复杂度： $O(n \log n)$ ，排序复杂度。

正确性： 反证，若最优队伍尾和贪心规则不一样，必然大于等于贪心规则，引出矛盾。

问题描述： n 个玩具建筑，每次操作可选择连续子区间使建筑高度 $+1$ ，求最少操作数使建筑从左到右单调递增。

输入： 长度为 n 的整形数组 a 。

输出： 最少操作数。

观察： 对区间 $[i, j]$ 上的操作，只改变 a_{i-1} 和 a_i , a_j 和 a_{j+1} 的相对值。目标为使得相邻两两高度差均为非正。每次操作至多向答案接近 1。

复杂度： $O(n)$

正确性： 从观察出发。

问题描述：一串只含A或B的字符串，子串AB/BB可消去，求最终剩下的最短长度。

输入：字符串s，只含A/B
输出：最终剩下最短字符串的长度

```
int main() {
    string s;
    cin >> s;
    int ans = 0;
    for(int i = 0; i < s.size(); i++){
        if(ans && s[i] == 'B') ans--;
        else ans++;
    }
    cout << ans << endl;
}
```

问题描述：对于含 n 个数的整数集合，判断其中是否有三个数，存在以它们为边长的非退化的三角形。

输入：集合大小 n 和集合中元素 a_1, \dots, a_n

输出：使得和最大的 k 分割，以及这样的分割的个数

```
int main() {
    int n;
    cin >> n;
    vector<long long> a(n);
    for(int i = 0; i < n; i++) cin >> a[i];
    if(n >= 100) {
        cout << "YES" << endl;
        return 0;
    }
    sort(a.begin(), a.end());
    for(int i = 1; i < n-1; i++) {
        if(a[i] + a[i-1] > a[i+1]) {
            cout << "YES" << endl;
            return 0;
        }
    }
    cout << "NO" << endl;
}
```

讨论