[05]贪心运用、数论运用

XY_cpp

2025-08-10

www.luogu.com.cn

课前须知

- 上课的时候专心听讲解，**不要跟着老师抄代码**，下课后独立完成。
- 不直接使用 AI 做题，AI 会做不等于自己会。
- 不抄袭题解（含对照题解抄一遍），抄对不等于会做。
  - 看完题解后，关闭题解独立练习。
  - 练习中途遇到问题，应当分析题目及自己的思路，而非回忆题解或再次参考题解。
- 做过的题在课后需要重新独立完成，不参考老师的课件、代码，不参考自己以前的代码。

贪心运用

贪心算法

贪心算法，是用计算机来模拟一个「贪心」的人做出决策的过程。这个人十分贪婪，每一步行动总是按某种指标选取最优的操作。而且他**目光短浅，总是只看眼前**，并不考虑以后可能造成的影响。

可想而知，并不是所有的时候贪心法都能获得最优解，所以一般使用贪心法的时候，都要确保自己能**证明其正确性**。

证明方法

- 反证法: **如果**交换方案中任意两个元素/相邻的两个元素后, 答案**不会变得更好**, 那么可以推定目前的解已经是最优解了。
- 归纳法：先算得出边界情况（例如 $n=1$）的最优解 F_1，然后再证明：对于每个n，F_{n+1} 都可以由 F_n 推导出结果。

例题-B4071

题目描述：

- 有m个范围是$[1,n]$的整数;
- 对于第i个数为p_i，可以花费c_i的代价将p_i改成$[1,n]$中的任何整数;
- 求将 1 的出现次数变得**唯一最大**的最小花费;

思路：

- 枚举 1 最终出现的次数，钦定为x次;
- 如果其他数字出现次数超出x次，就需要降至$x-1$，显然优先取花费代价小的，将其变为数字 1;
- 如果最后得到数字 1 的出现次数不足x，就在剩下的数字中挑取花费代价小的，将其变为数字 1;

例题-P11376

题目描述：

- 题目较长，需自行阅读并理解题意
- 点击跳转

例题-P11376

思路：

- 单独考虑第i辆货车在第k个站点，那么其开销为：

$$
\begin{aligned}
&2a_ip_k + 2b_i(x - p_k) \\
&= 2(a_i - b_i)p_k + 2b_ix
\end{aligned}
$$

- 贪心结论 1:
  - 若$a_i > b_i$，则p_k越小越好
  - 若$a_i < b_i$，则p_k越大越好
- 问题转换为哪辆货车**优先使用**站点

例题-P11376

- 贪心结论 2:
  - 当$a_i > b_i$时，$a_i - b_i$越大，就越应该使用p_k小的车站
  - 当$a_i < b_i$时，$a_i - b_i$越小，就越应该使用p_k大的车站
- 我们来证明$a_i > b_i$时的情况，按照这样的优先级分配时，考虑第j辆货车在第$l(k < l)$个站点，那么两者的开销为：

$$2(a_i - b_i)p_k + 2(a_j - b_j)p_l + 2(b_i + b_j)x$$

如果不按照这样的优先级分配，那么两者的开销为：

$$2(a_i - b_i)p_l + 2(a_j - b_j)p_k + 2(b_i + b_j)x$$

例题-P11376

后者减去前者，得到：

$$2\big((a_i-b_i)-(a_j-b_j)\big)(p_l-p_k)\geq 0$$

因此这样不会更优

- $a_i<b_i$时的情况同理可得

数论运用

例题-B3929

题目描述：

- 给定范围a和n个数x;
- 对于数x，寻找第一个不小于它的幸运数;
- 不小于a的**完全平方数**或**完全平方数的倍数**被称为幸运数;
- $1 \le n \le 2 \times 10^5, 1 \le a \le 10^6, 1 \le x \le 10^6 + 1;$

思路：

- 注意到最大的幸运数一定是$1001 * 1001$，因此可以用埃氏筛找到所有这个范围内的幸运数;
- 注意到x的范围也是固定的，因此可以提前预处理出所有的答案，然后每一次的寻找都是$O(1)$的;

例题-B3929

首先定义常量:

```
const int N = 1001 * 1001;
```

核心代码如下所示:

```
for (int i = ceil(sqrt(a)); i * i <= N; i++)
  for (int j = 1; j * i * i <= N; j++) {
    vis[j * i * i] = true;
  }
int lst;
for (int i = N; i >= 1; i--) {
  if (vis[i]) lst = i;
  ans[i] = lst;
}
```

例题-P10720

题目描述：

- 有n个数，求每一个数是否为幸运数字
- 对于一个数x，如果它恰好有两种不同的质因子，那么x就为幸运数字
- $1 \leq n \leq 10^4, 1 \leq x \leq 10^6$

思路：

- 直接拆分判断即可

最大公因数

最大公因数（Greatest Common Divisor, 简称 GCD），指两个或多个整数共有的最大因数。

通常我们使用$\gcd(a,b)$来表示两个数a,b之间的最大公因数，例如：

$$\gcd(12,18)=6$$

最大公因数可以使用**欧几里得辗转相除法**求得：

$$\gcd(a,b)=\begin{cases}\gcd(b,a-b) & \text{if } b\neq 0\\ a & \text{if } b=0\end{cases}$$

对应到代码实现：

```
int gcd(int a, int b) {
  if (b == 0) return a;
  return gcd(b, a % b);
}
```

例题-P13014

题目描述：

- 给定 n 个正整数 $a_1, a_2, \dots, a_n$，对于第$i(1 \le i \le q)$组询问，请求出$a_1 + i, a_2 + i, \dots, a_n + i$的最大公因数，也即$\gcd(a_1 + i, a_2 + i, \dots, a_n + i)$
- $1 \le n \le 10^5, 1 \le q \le 10^5, 1 \le a_i \le 1000$

思路：

- 考虑两个性质：
  - $\gcd(a, b) = \gcd(b, a - b)$
  - $\gcd(a, b, c) = \gcd(a, \gcd(b, c))$

例题-P13014

本质上所求最大公因数可以转换为如下形式：

$$\gcd(a_1 + i, \gcd(a_2 - a_1, a_3 - a_1, ..., a_n - a_1))$$

核心代码如下所示：

```cpp
int n, q;
cin >> n >> q;
int g = 0;
for (int i = 1; i <= n; i++) {
  cin >> a[i];
  if (i != 1) g = gcd(abs(a[i] - a[i - 1]), g);
}
for (int i = 1; i <= q; i++) cout << gcd(a[1] + i, g) << endl;
```

例题-P10719

题目描述：

- n行m列的网格图，其中每个格子要么是白色，要么是黑色
- 求：至少包含k个黑色格子的最小子矩形包含了多少个格子

思路：

- 使用 `sum[i][j]`记录黑色格点的二维前缀和
- 暴力枚举左上角(i,j)，右下角(k,l)
- 计算枚举出的矩形内的黑色格点数量

综合练习

例题-B4050

题目描述：

- t组数据，每组一个怪物血量h;
- 使用下面两种攻击将怪物血量刚好降至0:
  - 物理攻击：第i次攻击造成2^{i-1}点伤害;
  - 魔法攻击：任意选择一个质数x，造成x点伤害，最多用一次;
- 求攻击怪物的最少次数，无解输出-1;
- $t \le 10, h \le 10^5$;

思路：

- 暴力枚举进行了k次物理攻击，造成伤害为$2^k - 1$;
- 计算$x - 2^k + 1$是否为素数，如果是，进行魔法攻击;