贪心

NOIP历年真题

- 一个旅行家想驾驶汽车以最少的费用从一个城市到另一个城市（假设出发时油箱是空的）。
- 给定两个城市之间的距离、汽车油箱的容量、每升汽油能行驶的距离、出发点每升汽油价格P和沿途油站数 n 。
- 油站 i 离出发点的距离 d_i 、每升汽油价格 p_i 。
- 计算最小花费。
- $n \leq 100$

- 贪心策略就是选择去最近的比当前站便宜的加油站，若没有，就先加满，然后走到能到达的最便宜的加油站。

- luogu1016

- 有 n 个人，每个人有两个数 a_i, b_i 。
- 第 i 个人的得分为前 $i-1$ 个人的 a 的乘积再除以自己的 b_i 下取整的结果。
- 第1个人的位置固定，你可以自由安排其他人的位置。
- 最小化所有人的分数的最大值。

- a_i 表示左手数字，b_i 表示右手数字，w_i 表示应得金币。
- 考虑相邻两人谁在前更优，显然交换相邻两人对前面和后面都没有影响，设 p 表示之前所有人的 a 的乘积。
- x 在 y 之前：$w_x = \frac{p}{b_x}, w_y = \frac{p \times a_x}{b_y}$
- x 在 y 之后：$w_y = \frac{p}{b_y}, w_i = \frac{p \times a_y}{b_x}$
- 由于是使最大值最小，只需判断 $\frac{p \times a_x}{b_y}$ 与 $\frac{p \times a_y}{b_x}$ 的大小关系，即 $a_x \times b_x$ 与 $a_y \times b_y$ 的大小关系。
- 所以按 $a_i \times b_i$ 排序即可。

- luogu1080

- 约翰的工作日从0时刻开始，有 10^9 个单位时间。
- 在任一单位时间，他都可以选择编号1到 n 的 n 项工作中的任意一项工作来完成。
- 工作 i 的截止时间是 d_i，完成后获利是 p_i，只能完成一次。
- 在给定的工作利润和截止时间下，求约翰能够获得的利润最大为多少。
- $n \leq 10^5$
- 3
- 2 10
- 1 5
- 1 7
- 17

- 考虑每个时刻完成的工作。
- 遇到超时但收获更大的工作时，可以反悔。

- luogu2949

```
for (i = 1; i <= n; i++) {
  if (a[i].d <= (int)q.size()) {  // 超过截止时间
    if (q.top() < a[i].p) {       // 后悔
      ans += a[i].p - q.top();
      q.pop();
      q.push(a[i].p);
    }
  } else {  // 直接加入队列
    ans += a[i].p;
    q.push(a[i].p);
  }
}
```

- 给定n个数，多次询问选择k个数使和为奇数的最大和。n<=100000

- 先排序，若最大的k个数合法则直接选，否则可以将一个奇数替换为偶数或偶数替换为奇数。用最大的替换最小的即可。

- 有 n 个数字 $a_1, a_2, \ldots, a_n$ 排成一排，你需要从左到右依次越过所有数。
- 越过一个数的代价就是 a_i ，但你也可以选择跳过它，这样做的代价是0，但是之后越过其它数（不包括跳过）的代价会+1。
- 现在你拥有最多 k 次跳跃的机会，求最小的代价总和。
- $n \leq 10^5$
- 4 1
- 5 10 11 5
- 21

- 5+10+0+(5+1)=21

- 一定会用完所有 k 次跳跃：
- 若没有用完，跳最后一个越过的数。
- 设第一次跳跃位置为 i 。
- 首先会减少代价 a_i ，然后之后的越过都会+1，越过次数为 $n-i-(k-1)$，即后面数字数量-后面跳跃次数。
- 第 p 次跳跃减少的代价为 $a_i-(n-i)+(k-p)=a_i+i-n+k-p$ 。
- 后半部分与 i 无关，总和为 $\sum_{p=1}^{k}(-n+k-p)=\frac{k(k-1)}{2}-kn$ 。
- 最大化前半部分，即选择 k 个最大的 a_i+i 。

- cf1684D

- 有编号为1到 n 的 n 个点，你需要遍历所有点各一次。
- 从 i 号点到 j 号点的花费为 $(i+j)\ mod\ (n+1)$ 。
- 你可以任意选定起点，最小化花费之和。
- $n \le 10^5$

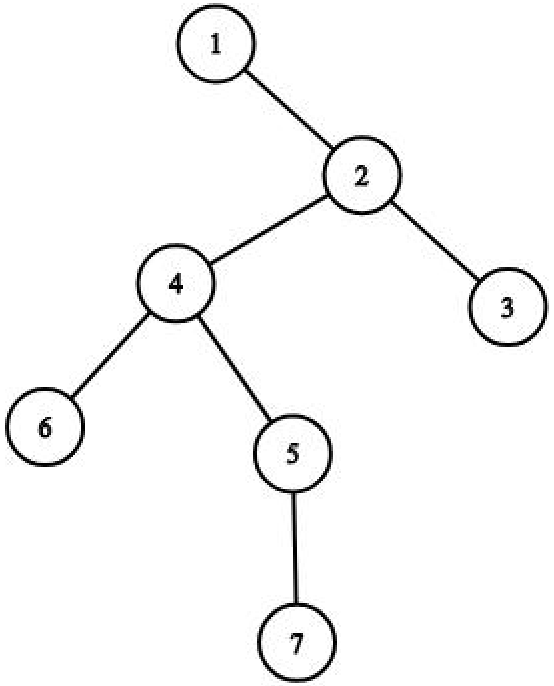
- $1 \to n \to 2 \to (n-1) \to 3 \to (n-2) \ldots$
- n 为奇数时答案为 $\frac{n-1}{2}$ 。
- n 为偶数时答案为 $\frac{n}{2} - 1$ 。

- cf804A

- 一张无向图，边权为0或1。你需要找出一棵生成树，使得边权为0的边的数量为一个给定值。

- 先优先加1做生成树，得到必须选的0边。这样选上这些0边后，任意再加一些0边达到给定值，一定能选出一些1边得到一棵生成树。

- 给定一棵树，求最少需要添加多少条边，使得所有节点到1号点的最短距离不超过2（最短路的边数）。
- $n \le 10^5$



- 每次考虑最深的点，连接其父节点与1号点。
- 用大根堆维护剩余点的深度，每次连边后将所有与其相邻的点删除。

- cf1029E

- 有一排 n 个数 a_i ，每次你可以选择两个相邻的数 x, y ，删去它们，然后用 $x - y$ 或者 $y - x$ 替换。
- $n - 1$ 次操作后会剩下一个数，求这个数的最大值。
- $n \leq 10^5$
- 4
- 2 1 2 1

- 2 -1 1
- 2 -2
- 4

- 最终的数字一定是 n 个数字各自乘以1或-1的和。
- 不可能出现的情况只有全为+1和全为-1。
- 也就是有 $2^n - 2$ 种不同情况。
- 归纳证明，对于 $n = 2$ 显然成立。
- 设对于 $n = k$ 成立，有 $2^k - 2$ 种情况，考虑在最后加一个数。
- 前 k 个数减去它和它减去前 k 个数是不同的，有 $2^{k+1} - 4$ 种。
- 前 k 个为+1，最后为-1和前 k 个为-1，最后为+1未考虑，可由第一个数和后 k 个数相减得到，故总数为 $2^{k+1} - 2$ 。
- 换句话说，至少会有一个+1和一个-1，我们让最大的数取+1，最小的数取-1，剩余的数取绝对值即可。
- cf1038D

- 有 n 个人想要坐车，线路可以抽象成一条数轴。
- 第 i 个人想要从坐标 s_i 坐到坐标 t_i 。
- 你的车从原点 0 出发，最终行驶到坐标 m 。
- 车上最多只能同时坐一个乘客，但你可以让乘客中途下车，只需保证最终将其送达他的目的地即可。
- 在满足所有人的需求下，你行驶的最小总路程是多少呢?

- $|s_i - t_i|$ 为必经路程，另对于每个 t_i 需要匹配一个 s_j 。
- 注意将 0 作为一个 t_0 ， m 作为一个 s_0 。
- 分别排序，按顺序匹配即可。

- 有 n 个物品，大小为 a_i ，你需要将其中的若干物品分为若干份，使得物品总数变为 $k(k \geq n)$ ，最小化分割后所有物品的大小的平方和。
- $n \leq k \leq 10^5$

- 3 6
- 5 3 1

- (1,2,2),(1,2),1
- 1+4+4+1+4+1=15

- 将一个物品 a_i 分成 p 份，要得到最小平方和，方法是将其尽量均分。
- 因此，一个物品分为 p 份的最小平方和容易计算。
- 初始状态为 n 个物品，每个都是1份。
- 对于每个物品，设其当前要被分为 p 份，那么将分为 p 份的平方和与分为 $p+1$ 份的平方和之差与其它物品进行比较。
- 每次取出最大的（差最大，也就是能减少最多的），多分一份，重新加入堆。
- 对于同一个物品，随着份数的增加，减少的平方和不断下降。

- cf1428E

- 一个有序的数组，将其分为 k 段，每段的价值就是段内的最大值减去最小值。
- 最小化所有段的价值之和。
- $n \leq 10^5$

- 6 3
- 4 8 15 16 23 42
- 12

- [4,8,15,16],[23],[42]

- 考虑差分数组。
- 分成 k 段相当于去掉了差分数组中的 $k-1$ 个数。
- 去掉最大的 $k-1$ 个即可。

- cf1197C

- 给定一个长为 n 的序列，选出 k 个长度在 $[l,r]$ 之间的子段，使它们的和最大。
- $n,k \leq 5 \times 10^5$

- 先以每个右端点 i 找出左端点在 $[i-r+1, i-l+1]$ 范围内的最大子段放入堆，每次找出最大的分裂成两部分重新加入堆。
- 用 ST 表维护区间内最小的左端点。

- bzoj2006

- n 个物品，重量为 w_i 。
- 你有一辆载重为 W 的车，每次你都会运走一些总重不超过 W 的物品，策略为不断选择能运的物品中最重的一个。
- 在运送总次数不超过 R 的前提下，求 W 的最小值。
- $R, n, w_i <= 2000$

- 直接二分答案并不对，因为答案不具有可二分性。
- 设最优解为 C ，显然 $W \geq C + \max\{w_i\}$ 时均合法，因为每次运送的总重量至少为 C ，若不足 C 会有超过 $\max\{w_i\}$ 的空余。
- 所以先二分出任一尽量小的合法解 T ，那么 $W < T - \max\{w_i\}$ 一定不合法，所以只需检验 $\max\{w_i\}$ 个解是否合法即可。
- $O(n^2 logn)$