

2024 年湖南多校训练-湘潭大学

方云卓, 付有鑫, 李子熙, 魏志高, 谢勇, 曾庆鑫, 卓越

计算机学院·网络空间安全学院

April 14, 2024

题目一览

题号	名称	作者	类型	难度
A	Average of Intervals	魏志高	动态规划	签到
B	Bigger, Bigger, Bigger	方云卓	动态规划	容易
C	Calculation of Intervals	魏志高	Hash	容易
D	DAY ⁻¹	李子熙	数学	容易
E	Election	李子熙	搜索	中
F	4 Rectangles and 1 Square	谢勇	暴力	签到
G	Grading Papers	付有鑫	字符串, 数据结构	难
H	HNOI2010	李子熙	图论, 数据结构	难
I	Intervals	曾庆鑫	思维, 数学	容易
J	Journey on the Number Line	魏志高	动态规划	中
K	Kitchen	卓越	贪心	签到

思路

- 通过平均数的性质可以发现，最大权值即为最大子段和。原题转化为求序列的最大子段和以及最多可以取几个区间，使得每个区间的和都等于最大子段和。
- 最大子段和可以通过动态规划解决，而最多区间数量只需在动规过程中额外特判转移。如果以第 i 位结尾的区间最大子段和等于整体最大子段和。那么答案加一并且 $dp[i + 1]$ 不能从 $dp[i]$ 继承转移。相当于直接将 i 结尾的最大子段和区间直接取走。容易证明，这样贪心显然是对的。
- 时间复杂度 $\mathcal{O}(n)$

A. Average of Intervals

```
1  for(int i=1;i<=n;i++)
2  {
3      cin>>arr[i];
4      ban[i]=false;
5      dp[i]=arr[i]+max(0ll,dp[i-1]);
6      ans=max(ans,dp[i]);
7  }
8  for(int i=1;i<=n;i++)
9  {
10     dp[i]=arr[i];
11     if(!ban[i-1])
12     {
13         dp[i]+=max(0ll,dp[i-1]);
14     }
15     if(dp[i]==ans)
16     {
17         ban[i]=true;
18         cnt++;
19     }
20 }
```

K. Kitchen

思路

- 枚举所有可能产生的猪脚饭，如果其权值在原给定序列的最大值和最小值之间则插入给定序列
- 暴力枚举检查即可。

K. Kitchen

```
1  for(int i=1;i<=x;i++)
2      for(int j=1;j<=y;j++)
3          if(u[i]*v[j]<=mx&&u[i]*v[j]>=mi)
4              b[u[i]*v[j]]=1;
5  int now=1,ans=0;
6  for(int i=mi+1;i<=mx;i++)
7  {
8      if(!b[i])now++;
9      else
10         {
11             ans=max(ans,now);
12             now=1;
13         }
14  }
15  cout<<ans;
```

思路

- 先考虑，两个矩形合并成一个矩形，枚举长宽组合最多产生五种结果（包括无法组成）
- 对于三个矩形，前两个合并成的矩形，再和第三个矩形合并
- 对于四个矩形，枚举四个矩形的顺序
- 前两个合并成的矩形，和后两个形成三个矩形的情况（有两种，即新矩形放前面和放后面）
- 码农题，标程写的力大砖飞。此题也可以进一步扩展到 n ，不过需要写个状态压缩的枚举递归。

B. Bigger, Bigger, Bigger

思路

- 注意, dp_1 与 dp_2 的初始状态不同。由于 $p_{T-1} < \sqrt{z}$, 设操作次数上限为 $\max t$, 我们只需预处理出 $1 \leq i < \sqrt{z}$, $1 \leq j \leq \max t$ 时的 dp_1 与 dp_2 。先从小到大枚举 j , 然后枚举 p_{T-1} , 若 $p_{T-1} \cdot dp[p_{T-1}][j] < z$, 且 $(p_{T-1} \cdot k) \cdot (dp[p_{T-1}][j] + d) \geq z$ 或 $(p_{T-1} + d) \cdot (dp[p_{T-1}][j] \cdot k) \geq z$, 即为合法状态, 更新答案为 $\text{ans} = j + 1$ 。
- 复杂度为 $\mathcal{O}(\sqrt{z} \cdot \log(z))$ 。

思路

- 考虑怎么计算满足“区间内每一个数的出现次数都为偶数”
- 将原值域内的每一个数随机映射到 $[0, V]$ 这个区间，一个区间满足条件有 $\frac{V-1}{V}$ 的概率等价于该区间异或和为 0。
- 整体错误率为 $\frac{n^2}{V}$ ，取 $V = 10^{18}$ 时错误概率不超过 $\frac{1}{10^6}$ 。

思路

- 根据鸽巢原理, $n > 31$ 时答案必为 "NO"。
- 容易发现问题是在问能否选择一个非空子集, 使得在每个数前添 " +/ - " 号后和为 0。
- 使用折半搜索即可。

J. Journey on the Number Line

思路

- 先将给定的 n 个点按照 x 坐标从小到大排序
- 令 $dp[i][j]$ 表示考虑完按照 x 值从小到大排序后长度为 i 的前缀，当前答案形成了 j 个连通块的最小代价。
- 若 i 不为起点或终点，则 $dp[i][j]$ 可以从 $dp[i-1][j-1]$ 和 $dp[i-1][j]$ 以及 $dp[i-1][j+1]$ 转移过来。其对应的意义分别为将第 i 个点当成一个独立的连通块插入，将第 i 个点并入已有的连通块，用第 i 个点合并已有的连通块。
- 若 i 是起点或者终点，则 $dp[i][j]$ 可以从 $dp[i-1][j-1]$ 和 $dp[i-1][j]$ 转移过来。需要注意的是，这种情况下第 i 个点只能产生单边贡献。
- 将 $dp[i][j]$ 开成 pair 数组即可在转移的过程中记录方案数。答案即为 $dp[n][1]$ 。
- 时间复杂度为 $\mathcal{O}(n^2)$

G. Grading Papers

思路

- 首先考虑哈希所有的字符串前缀，并对每个哈希值，存下哪些位置拥有这个哈希值。
- 对于每个询问串，同时也将其哈希，那么在不考虑带修的情况下可以做到 $\mathcal{O}(q \log(n))$ 。
- 考虑带修，显然不能对于每次修改都重构一次该串的哈希，如果我们每隔 B 次查询就重构一次所有串的哈希值，那么重构的整体复杂度为 $\mathcal{O}(nq/B)$ ，然后对于当前重构到下次重构之间的询问，暴力计算每次修改对询问串给定的指定长度的哈希值的影响。算完之后再还原，这样的询问的复杂度为 $\mathcal{O}(qB)$ ， B 取一个合适的数字便可通过。

提示

- 标程偷懒重构使用了 `unordered_map` 作为哈希位置映射，此时重构的常数略大，需要 B 取大一点，但通过时限已经绰绰有余了。
- 比赛中有队伍使用对串长分块通过了 (比标程更快)，具体点 [这里](#)。

思路

- 远古省选题。
- 可以考虑用并查集做二分图判定，我们将区间按其中一个端点排序，然后依次枚举，可以发现每次会在线段树中找到一些区间与当前区间相交。可以发现每次相交以后所有区间和成了一个连通块，于是我们可以只保留这个连通块。
- 复杂度均摊是 $\mathcal{O}(n \log_2 n)$ 。

