SUSTechCPC 2021

Solutions presentation

December 25, 2021

A: Ah, Tenshi!

Problem Author: FluffyBunny

问题: 给定长度为 n 数组和一个初始等级值，可以按照任意顺序选择数组内数个值，如果当前等级比选择值大就能加上选择的值，求最多到多少级。

A: Ah, Tenshi!

Problem Author: FluffyBunny

问题: 给定长度为 n 数组和一个初始等级值，可以按照任意顺序选择数组内数个值，如果当前等级比选择值大就能加上选择的值，求最多到多少级。

考虑到如果当前等级小于等于当前数组中的最小值，对于数组中的任意值，我们都无法选择。

A: Ah, Tenshi!

Problem Author: FluffyBunny

Solution

把数组升序排序，顺序扫一遍，如果当前等级比当前值大就加上当前值，否则直接输出答案。

A: Ah, Tenshi!

Problem Author: FluffyBunny

Solution

把数组升序排序，顺序扫一遍，如果当前等级比当前值大就加上当前值，否则直接输出答案。

复杂度 $O(nlogn)$

A: Ah, Tenshi!

Problem Author: FluffyBunny

Solution

把数组升序排序，顺序扫一遍，如果当前等级比当前值大就加上当前值，否则直接输出答案。

复杂度 $O(nlogn)$

代码长度：363 B 用时：0.122 s

B: Battle of Poilogtopia

Problem Author: FluffyBunny

问题: 给个无相带权图，还有一个边的序列，A 和 B 玩游戏，A 可以选择边序列的某个前缀以某种代价加进图里，图确定下来之后 B 可以选择一些点。B 的分数是他选的点权，A 的分数是所有满足连着的两个点恰好有一个点被 B 选中的边权和-A 添加边花费的代价。两个人都希望自己的分数-对方的分数最大。问你最终的分数。

B: Battle of Poilogtopia

Problem Author: FluffyBunny

问题: 给个无相带权图，还有一个边的序列，A 和 B 玩游戏，A 可以选择边序列的某个前缀以某种代价加进图里，图确定下来之后 B 可以选择一些点。B 的分数是他选的点权，A 的分数是所有满足连着的两个点恰好有一个点被 B 选中的边权和-A 添加边花费的代价。两个人都希望自己的分数-对方的分数最大。问你最终的分数。

如果 A 没有办法修改图怎么做？

B: Battle of Poilogtopia

Problem Author: FluffyBunny

问题: 给个无相带权图，还有一个边的序列，A 和 B 玩游戏，A 可以选择边序列的某个前缀以某种代价加进图里，图确定下来之后 B 可以选择一些点。B 的分数是他选的点权，A 的分数是所有满足连着的两个点恰好有一个点被 B 选中的边权和-A 添加边花费的代价。两个人都希望自己的分数-对方的分数最大。问你最终的分数。

如果 A 没有办法修改图怎么做？

考虑 B 的方案，如果 B 选择了某个点会贡献点权的代价，如果 B 选择了某条边恰好某一端的点，就会贡献-边权的代价。

B: Battle of Poilogtopia

Problem Author: FluffyBunny

问题: 给个无相带权图，还有一个边的序列，A 和 B 玩游戏，A 可以选择边序列的某个前缀以某种代价加进图里，图确定下来之后 B 可以选择一些点。B 的分数是他选的点权，A 的分数是所有满足连着的两个点恰好有一个点被 B 选中的边权和-A 添加边花费的代价。两个人都希望自己的分数-对方的分数最大。问你最终的分数。

如果 A 没有办法修改图怎么做？

考虑 B 的方案，如果 B 选择了某个点会贡献点权的代价，如果 B 选择了某条边恰好某一端的点，就会贡献-边权的代价。

考虑网络流模型

B: Battle of Poilogtopia

Problem Author: FluffyBunny

Solution

考虑网络流最小割对于 B 来说每个点要么选要么不选，即 $x \in S$ 或 $x \in T$，S 代表选的点，T 代表不选的点。

B: Battle of Poilogtopia

Problem Author: FluffyBunny

Solution

考虑网络流最小割对于 B 来说每个点要么选要么不选，即 $x \in S$ 或 $x \in T$，S 代表选的点，T 代表不选的点。

每个点如果是正权就和 S 连一条边，负权和 T 连一条边。对于每条边，只需要把对应的点双向都连边就行了，这样一个选一个不选必定会割掉中间的边。由于是最小割，B 的最优选择下就是当前答案-网络流贡献

B: Battle of Poilogtopia

Problem Author: FluffyBunny

Solution

考虑网络流最小割对于 B 来说每个点要么选要么不选，即 $x \in S$ 或 $x \in T$，S 代表选的点，T 代表不选的点。

每个点如果是正权就和 S 连一条边，负权和 T 连一条边。对于每条边，只需要把对应的点双向都连边就行了，这样一个选一个不选必定会割掉中间的边。由于是最小割，B 的最优选择下就是当前答案-网络流贡献

加上修改考虑 A 的选择，每次其实是往图中加了一条边，直接在残余网络上加边跑就行了。最后的答案取最小值。

B: Battle of Poilogtopia

Problem Author: FluffyBunny

Solution

考虑网络流最小割对于 B 来说每个点要么选要么不选，即 $x \in S$ 或 $x \in T$，S 代表选的点，T 代表不选的点。

每个点如果是正权就和 S 连一条边，负权和 T 连一条边。对于每条边，只需要把对应的点双向都连边就行了，这样一个选一个不选必定会割掉中间的边。由于是最小割，B 的最优选择下就是当前答案-网络流贡献

加上修改考虑 A 的选择，每次其实是往图中加了一条边，直接在残余网络上加边跑就行了。最后的答案取最小值。

复杂度 $O(N^2MK)$ 这是理论复杂度上界，但实际上根本跑不满。

B: Battle of Poilogtopia

Problem Author: FluffyBunny

Solution

考虑网络流最小割对于 B 来说每个点要么选要么不选，即 $x \in S$ 或 $x \in T$，S 代表选的点，T 代表不选的点。

每个点如果是正权就和 S 连一条边，负权和 T 连一条边。对于每条边，只需要把对应的点双向都连边就行了，这样一个选一个不选必定会割掉中间的边。由于是最小割，B 的最优选择下就是当前答案-网络流贡献

加上修改考虑 A 的选择，每次其实是往图中加了一条边，直接在残余网络上加边跑就行了。最后的答案取最小值。

复杂度 $O(N^2MK)$ 这是理论复杂度上界，但实际上根本跑不满。

代码长度：1786 B 用时：0.128 s

C: Cknight and String

Problem Author: Satori

问题: 给定仅由字符 ab 组成的字符串 s，问最小修改字符个数使得不存在子串“ab”

C: Cknight and String

Problem Author: Satori

问题: 给定仅由字符 ab 组成的字符串 s，问最小修改字符个数使得不存在子串“ab”

需要 s 中不存在子串“ab”，只需枚举分界线，使得分界线前都是字符'b'，分界线后都是字符'a'. 取修改次数最小值即可

C: Cknight and String

Problem Author: Satori

问题: 给定仅由字符 ab 组成的字符串 s，问最小修改字符个数使得不存在子串“ab”

需要 s 中不存在子串“ab”，只需枚举分界线，使得分界线前都是字符'b'，分界线后都是字符'a'. 取修改次数最小值即可

复杂度:$O(n)$

代码长度：329 用时：

D: DengSH Loves Kebab

Problem Author: Satori

问题: 给定 N 个长度为 M 的字符串，每次等概率随机抽取一个，问至少知道前多少位可以确定字符串是哪一个。

D: DengSH Loves Kebab

Problem Author: Satori

问题: 给定 N 个长度为 M 的字符串，每次等概率随机抽取一个，问至少知道前多少位可以确定字符串是哪一个。

对于每个字符串，当知道它和其他所有串的最大的 LCP+1 位时可以确定。

D: DengSH Loves Kebab

Problem Author: Satori

问题: 给定 N 个长度为 M 的字符串，每次等概率随机抽取一个，问至少知道前多少位可以确定字符串是哪一个。

对于每个字符串，当知道它和其他所有串的最大的 LCP+1 位时可以确定。

做法一: 对于每个字符串枚举其他所有字符串算 LCP 取最大值。时间复杂度 $O(n^2m)$

D: DengSH Loves Kebab

Problem Author: Satori

问题: 给定 N 个长度为 M 的字符串，每次等概率随机抽取一个，问至少知道前多少位可以确定字符串是哪一个。

对于每个字符串，当知道它和其他所有串的最大的 LCP+1 位时可以确定。

做法一: 对于每个字符串枚举其他所有字符串算 LCP 取最大值。时间复杂度 $O(n^2m)$

做法二: 按字典序排序后，LCP 只需要算相邻的两个。时间复杂度 $O(nmlogn)$

D: DengSH Loves Kebab

Problem Author: Satori

问题: 给定 N 个长度为 M 的字符串，每次等概率随机抽取一个，问至少知道前多少位可以确定字符串是哪一个。

对于每个字符串，当知道它和其他所有串的最大的 LCP+1 位时可以确定。

做法一: 对于每个字符串枚举其他所有字符串算 LCP 取最大值。时间复杂度 $O(n^2m)$

做法二: 按字典序排序后，LCP 只需要算相邻的两个。时间复杂度 $O(nmlogn)$

做法三: 建 trie 树后直接算和其他串的 LCP。时间复杂度 $O(nm)$

E: Eager for Tea

Problem Author: TmofZD

- **题目:** 询问期望随机选择几次 $[0, x]$ 中的实数进行累加后，结果不小于 1。

E: Eager for Tea

Problem Author: TmofZD

- **题目:** 询问期望随机选择几次 $[0, x]$ 中的实数进行累加后，结果不小于 1。
- **答案:** $e^{\frac{1}{x}}$

E: Eager for Tea

Problem Author: TmofZD

- **题目:** 询问期望随机选择几次 $[0, x]$ 中的实数进行累加后，结果不小于 1。
- **答案:** $e^{\frac{1}{x}}$
- **解答:** 期望为 $\sum_{i=1}^{\infty} i * ((\frac{1}{x})^i * (1 - \frac{1}{i}) * (\frac{1}{(i-1)!}) + (\frac{1}{x})^{i-1} * (1 - \frac{1}{x}) * \frac{1}{(i-1)!})$

F: Fix a Weird Speaker

Problem Author: RogerDTZ

问题: 给定文本串 S 和模式串 T，求 S 的一个最长子串，使得其为 T 重复其某个子区间的结果。

F: Fix a Weird Speaker

Problem Author: RogerDTZ

问题: 给定文本串 S 和模式串 T，求 S 的一个最长子串，使得其为 T 重复其某个子区间的结果。

KMP

F: Fix a Weird Speaker

Problem Author: RogerDTZ

问题: 给定文本串 S 和模式串 T，求 S 的一个最长子串，使得其为 T 重复其某个子区间的结果。

KMP

对 S 正向做一次 T 的匹配，记录 S 每个前缀 pre_i 尾部能匹配到的 T 最长的前缀，记做 f_i。

F: Fix a Weird Speaker

Problem Author: RogerDTZ

问题: 给定文本串 S 和模式串 T，求 S 的一个最长子串，使得其为 T 重复其某个子区间的结果。

KMP

对 S 正向做一次 T 的匹配，记录 S 每个前缀 pre_i 尾部能匹配到的 T 最长的前缀，记做 f_i。

对 S 逆向做一次 T 的匹配，记录 S 每个后缀 suf_i 前部能匹配到的 T 最长的后缀，记做 g_i。

F: Fix a Weird Speaker

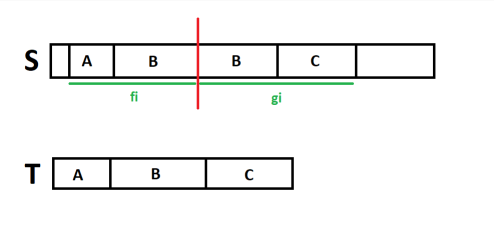
Problem Author: RogerDTZ

问题: 给定文本串 S 和模式串 T，求 S 的一个最长子串，使得其为 T 重复其某个子区间的结果。

KMP

对 S 正向做一次 T 的匹配，记录 S 每个前缀 pre_i 尾部能匹配到的 T 最长的前缀，记做 f_i。

对 S 逆向做一次 T 的匹配，记录 S 每个后缀 suf_i 前部能匹配到的 T 最长的后缀，记做 g_i。



F: Fix a Weird Speaker

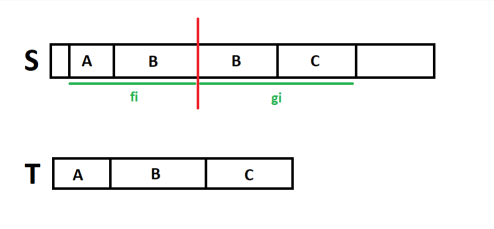
Problem Author: RogerDTZ

问题: 给定文本串 S 和模式串 T，求 S 的一个最长子串，使得其为 T 重复其某个子区间的结果。

KMP

对 S 正向做一次 T 的匹配，记录 S 每个前缀 pre_i 尾部能匹配到的 T 最长的前缀，记做 f_i。

对 S 逆向做一次 T 的匹配，记录 S 每个后缀 suf_i 前部能匹配到的 T 最长的后缀，记做 g_i。



如果 $f_i + g_{i+1} > |T|$，则可以更新答案。

F: Fix a Weird Speaker

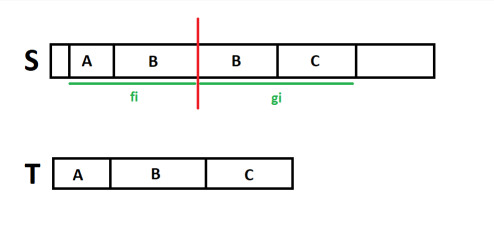
Problem Author: RogerDTZ

问题: 给定文本串 S 和模式串 T，求 S 的一个最长子串，使得其为 T 重复其某个子区间的结果。

KMP

对 S 正向做一次 T 的匹配，记录 S 每个前缀 pre_i 尾部能匹配到的 T 最长的前缀，记做 f_i。

对 S 逆向做一次 T 的匹配，记录 S 每个后缀 suf_i 前部能匹配到的 T 最长的后缀，记做 g_i。



如果 $f_i + g_{i+1} > |T|$，则可以更新答案。

时间复杂度：$O(|S| + |T|)$

G: Gorgeous Andrea

Problem Author: Satori

问题: 给定长度为 n 数组 a,b,c，求最大非空子序列使得 $\sum_{i=1}^{k} c_{p_i} + \sum_{i=2}^{k} a_{p_i} \cdot b_{p_{i-1}}$ 最大（p 表示子序列）

G: Gorgeous Andrea

Problem Author: Satori

问题: 给定长度为 n 数组 a,b,c，求最大非空子序列使得 $\sum_{i=1}^{k} c_{p_i} + \sum_{i=2}^{k} a_{p_i} \cdot b_{p_{i-1}}$ 最大（p 表示子序列）

用 dp[i] 表示已经考虑了石子 1...i，并且选择了石子 i 的最大开心值，有转移方程

$$dp[i] = max\{dp[j] + a[i] * b[j]\} + c[i](0 \leq j < i)$$

G: Gorgeous Andrea

Problem Author: Satori

问题: 给定长度为 n 数组 a,b,c，求最大非空子序列使得 $\sum_{i=1}^{k} c_{p_i} + \sum_{i=2}^{k} a_{p_i} \cdot b_{p_{i-1}}$ 最大（p 表示子序列）

用 dp[i] 表示已经考虑了石子 1...i，并且选择了石子 i 的最大开心值，有转移方程

$$dp[i] = max\{dp[j] + a[i] * b[j]\} + c[i](0 \leq j < i)$$

可以发现可以用斜率优化

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

分治时每次先处理完左半支，然后用左半支已知的 dp 值造凸包，更新右半支的 dp 值。

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

分治时每次先处理完左半支，然后用左半支已知的 dp 值造凸包，更新右半支的 dp 值。

复杂度 $O(nlog^2n)$

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

分治时每次先处理完左半支，然后用左半支已知的 dp 值造凸包，更新右半支的 dp 值。

复杂度 $O(nlog^2n)$

需要注意的是建造凸包的过程中乘法可能会爆 longlong，需要使用 int128 改成除法 double 高精度解决。

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

分治时每次先处理完左半支，然后用左半支已知的 dp 值造凸包，更新右半支的 dp 值。

复杂度 $O(nlog^2n)$

需要注意的是建造凸包的过程中乘法可能会爆 longlong，需要使用 int128 改成除法 double 高精度解决。

代码长度：1455 B 用时：0.630 s

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

分治时每次先处理完左半支，然后用左半支已知的 dp 值造凸包，更新右半支的 dp 值。

复杂度 $O(nlog^2n)$

需要注意的是建造凸包的过程中乘法可能会爆 longlong，需要使用 int128 改成除法 double 高精度解决。

代码长度：1455 B 用时：0.630 s

Solution 2

标记永久化线段树，离散化 a[i]，之后不断在线段树上加边 + 查询。

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

分治时每次先处理完左半支，然后用左半支已知的 dp 值造凸包，更新右半支的 dp 值。

复杂度 $O(nlog^2n)$

需要注意的是建造凸包的过程中乘法可能会爆 longlong，需要使用 int128 改成除法 double 高精度解决。

代码长度：1455 B 用时：0.630 s

Solution 2

标记永久化线段树，离散化 a[i]，之后不断在线段树上加边 + 查询。

复杂度 $O(nlogn)$

G: Gorgeous Andrea

Problem Author: Satori

Solution 1

CDQ 分治 + 斜率优化

分治时每次先处理完左半支，然后用左半支已知的 dp 值造凸包，更新右半支的 dp 值。

复杂度 $O(nlog^2n)$

需要注意的是建造凸包的过程中乘法可能会爆 longlong，需要使用 int128 改成除法 double 高精度解决。

代码长度：1455 B 用时：0.630 s

Solution 2

标记永久化线段树，离散化 a[i]，之后不断在线段树上加边 + 查询。

复杂度 $O(nlogn)$

代码长度：1602 B 用时：0.180 s

H: Here Comes Chao Man

Problem Author: RogerDTZ

问题: 给定序列 a_i 和常数 E，问有多少种将其划分为若干非空段的方案，使得相邻两段的最小值至少相差 E。

H: Here Comes Chao Man

Problem Author: RogerDTZ

问题: 给定序列 a_i 和常数 E，问有多少种将其划分为若干非空段的方案，使得相邻两段的最小值至少相差 E。

设 f_i 表示，已经考虑 $a_1 \dots a_i$ 的划分，且 a_i 为其所在段的首个最小值，有多少合法划分方案（a_i 所在段可能还未闭合）。

H: Here Comes Chao Man

Problem Author: RogerDTZ

问题: 给定序列 a_i 和常数 E，问有多少种将其划分为若干非空段的方案，使得相邻两段的最小值至少相差 E。

设 f_i 表示，已经考虑 $a_1 \dots a_i$ 的划分，且 a_i 为其所在段的首个最小值，有多少合法划分方案（a_i 所在段可能还未闭合）。

考虑对于 i，哪些区间 $[l, r]$ 满足 a_i 是首个最小值

H: Here Comes Chao Man

Problem Author: RogerDTZ

问题: 给定序列 a_i 和常数 E，问有多少种将其划分为若干非空段的方案，使得相邻两段的最小值至少相差 E。

设 f_i 表示，已经考虑 $a_1 \dots a_i$ 的划分，且 a_i 为其所在段的首个最小值，有多少合法划分方案（a_i 所在段可能还未闭合）。

考虑对于 i，哪些区间 $[l, r]$ 满足 a_i 是首个最小值

用单调栈预处理 i 左侧第一个 $\leq a_i$ 的位置 L_i，以及 i 右侧第一个 $< a_i$ 的位置 R_i

H: Here Comes Chao Man

Problem Author: RogerDTZ

问题: 给定序列 a_i 和常数 E，问有多少种将其划分为若干非空段的方案，使得相邻两段的最小值至少相差 E。

设 f_i 表示，已经考虑 $a_1 \dots a_i$ 的划分，且 a_i 为其所在段的首个最小值，有多少合法划分方案（a_i 所在段可能还未闭合）。

考虑对于 i，哪些区间 $[l, r]$ 满足 a_i 是首个最小值

用单调栈预处理 i 左侧第一个 $\leq a_i$ 的位置 L_i，以及 i 右侧第一个 $< a_i$ 的位置 R_i

则 $L_i < l \leq r < R_i$ 满足 a_i 为其首个最小值

H: Here Comes Chao Man

Problem Author: RogerDTZ

问题: 给定序列 a_i 和常数 E，问有多少种将其划分为若干非空段的方案，使得相邻两段的最小值至少相差 E。

设 f_i 表示，已经考虑 $a_1 \dots a_i$ 的划分，且 a_i 为其所在段的首个最小值，有多少合法划分方案（a_i 所在段可能还未闭合）。

考虑对于 i，哪些区间 $[l, r]$ 满足 a_i 是首个最小值

用单调栈预处理 i 左侧第一个 $\leq a_i$ 的位置 L_i，以及 i 右侧第一个 $< a_i$ 的位置 R_i

则 $L_i < l \leq r < R_i$ 满足 a_i 为其首个最小值

考虑哪些 j ($j < i$) 可以用 f_j 贡献 f_i

1. $a_j \leq a_i - E$
2. $a_j \geq a_i + E$

H: Here Comes Chao Man

Problem Author: RogerDTZ

第一类: $a_j \leq a_i - E$

考虑 j 和 i 之间有多少种可能的断点

H: Here Comes Chao Man

Problem Author: RogerDTZ

第一类: $a_j \leq a_i - E$

考虑 j 和 i 之间有多少种可能的断点

1. 依据我们对 i 成为最小值的区间的观察，断点必须在 L_i 右边
   又注意到这些 j 显然满足 $j \leq L_i$
   故 $(L_i, i]$ 都可成为 j 所在段与 i 所在段的断点

H: Here Comes Chao Man

Problem Author: RogerDTZ

第一类: $a_j \leq a_i - E$

考虑 j 和 i 之间有多少种可能的断点

1. 依据我们对 i 成为最小值的区间的观察，断点必须在 L_i 右边
   又注意到这些 j 显然满足 $j \leq L_i$
   故 $(L_i, i]$ 都可成为 j 所在段与 i 所在段的断点
2. 对于一个 j，它在 $i \in (j, R_j)$ 可能成为一个能对 i 做贡献的 j

H: Here Comes Chao Man

Problem Author: RogerDTZ

第一类： $a_j \leq a_i - E$

考虑 j 和 i 之间有多少种可能的断点

1. 依据我们对 i 成为最小值的区间的观察，断点必须在 L_i 右边
   又注意到这些 j 显然满足 $j \leq L_i$
   故 $(L_i, i]$ 都可成为 j 所在段与 i 所在段的断点
2. 对于一个 j，它在 $i \in (j, R_j)$ 可能成为一个能对 i 做贡献的 j

使用树状数组，维护对于目前 i 可贡献的那些 j，以 a_j 为下标的 f 的前缀和

$$f_i \leftarrow^{+} (i - L_i) * bit.query(a_i - E)$$

H: Here Comes Chao Man

Problem Author: RogerDTZ

第二类: $a_j \geq a_i + E$

考虑 j 和 i 之间有多少种可能的断点

H: Here Comes Chao Man

Problem Author: RogerDTZ

第二类: $a_j \geq a_i + E$

考虑 j 和 i 之间有多少种可能的断点

1. 依据我们对 j 成为最小值的区间的观察，断点必须在 R_j 左边
   又注意到 $i \geq R_j$
   故 $(j, R_j]$ 都可以成为 j 所在段和 i 所在段的断点

H: Here Comes Chao Man

Problem Author: RogerDTZ

第二类： $a_j \geq a_i + E$

考虑 j 和 i 之间有多少种可能的断点

1. 依据我们对 j 成为最小值的区间的观察，断点必须在 R_j 左边
   又注意到 $i \geq R_j$
   故 $(j, R_j]$ 都可以成为 j 所在段和 i 所在段的断点
2. a_i 必须是断点到 i 这一段的最小值
   则 f_i 从第二类 j 获得的**实际贡献** $contr_i$，应为满足要求 1. 的所有 j 对其的贡献之和 $\sum_j f_j \times (R_j - j)$，减去 $contr_{L_i}$
   如此一来便能抵消所有 a_i 并非断点到 i 这一段的最小值的情况下的非法贡献。

H: Here Comes Chao Man

Problem Author: RogerDTZ

第二类： $a_j \geq a_i + E$

考虑 j 和 i 之间有多少种可能的断点

1. 依据我们对 j 成为最小值的区间的观察，断点必须在 R_j 左边
   又注意到 $i \geq R_j$
   故 $(j, R_j]$ 都可以成为 j 所在段和 i 所在段的断点
2. a_i 必须是断点到 i 这一段的最小值
   则 f_i 从第二类 j 获得的**实际贡献** $contr_i$，应为满足要求 1. 的所有 j 对其的贡献之和 $\sum_j f_j \times (R_j - j)$，减去 $contr_{L_i}$
   如此一来便能抵消所有 a_i 并非断点到 i 这一段的最小值的情况下的非法贡献。

计算完 f_i 后，令 $contr_k \leftarrow^+ f_i \ (k \geq l)$，其中 l 为最小的满足 $l > i$ 且 $a_l \leq a_i - E$ 的下标

同时令 $L_j = i$ 的 j 的 $contr_i$ 减去自己的 $contr_i$

$$f_i \leftarrow^+ contr_i$$

H: Here Comes Chao Man

Problem Author: RogerDTZ

时间复杂度 $O(N \log N)$

耗时 0.4 s 代码长度 3077 B

I: Invasion of Sjkmost

Problem Author: FluffyBunny

问题: 给个网格，每个格子是 0 或 1，你至少需要把多少 0 变成 1 才能通过走 1 的四联通从网格一边走到另一边？

I: Invasion of Sjkmost

Problem Author: FluffyBunny

问题: 给个网格，每个格子是 0 或 1，你至少需要把多少 0 变成 1 才能通过走 1 的四联通从网格一边走到另一边？

每个点可以和他的相邻的点连一条边

I: Invasion of Sjkmost

Problem Author: FluffyBunny

问题: 给个网格，每个格子是 0 或 1，你至少需要把多少 0 变成 1 才能通过走 1 的四联通从网格一边走到另一边？

每个点可以和他的相邻的点连一条边

考虑最短路

I: Invasion of Sjkmost

Problem Author: FluffyBunny

问题: 给个网格，每个格子是 0 或 1，你至少需要把多少 0 变成 1 才能通过走 1 的四联通从网格一边走到另一边？

每个点可以和他的相邻的点连一条边

考虑最短路

如果某个点是 1，那么到他不需要代价，否则需要 1 的代价（把这个点变成 1）

I: Invasion of Sjkmost

Problem Author: FluffyBunny

问题: 给个网格，每个格子是 0 或 1，你至少需要把多少 0 变成 1 才能通过走 1 的四联通从网格一边走到另一边？

每个点可以和他的相邻的点连一条边

考虑最短路

如果某个点是 1，那么到他不需要代价，否则需要 1 的代价（把这个点变成 1）

添加一个起始点，和第一行的点连边，添加一个终点，和最后一行的点连边

I: Invasion of Sjkmost

Problem Author: FluffyBunny

Solution 01bfs

I: Invasion of Sjkmost

Problem Author: FluffyBunny

Solution 01bfs

开个双端队列存当前访问的点，如果当前边权是 0，往队列头添加这个点，否则往队列尾添加这个点
这样每个点只会进出队列 $O(1)$ 次

I: Invasion of Sjkmost

Problem Author: FluffyBunny

Solution 01bfs

开个双端队列存当前访问的点，如果当前边权是 0，往队列头添加这个点，否则往队列尾添加这个点
这样每个点只会进出队列 $O(1)$ 次

复杂度 $O(NM)$

I: Invasion of Sjkmost

Problem Author: FluffyBunny

Solution 01bfs

开个双端队列存当前访问的点，如果当前边权是 0，往队列头添加这个点，否则往队列尾添加这个点
这样每个点只会进出队列 $O(1)$ 次

复杂度 $O(NM)$

代码长度：1261 B 用时：0.531 s

J: Join TB's Party

Problem Author: RogerDTZ

问题: 给定一棵边带权树，有 M 人从根节点出发，去各节点拿一把椅子回到根节点。每个人经过一条边需要一定时间，每个节点提供的椅子有数量上限 C_i，每个节点提供两把椅子的时间至少相差 P_i。问所有人拿一把椅子回到根节点至少需要多少时间。

J: Join TB's Party

Problem Author: RogerDTZ

问题: 给定一棵边带权树，有 M 人从根节点出发，去各节点拿一把椅子回到根节点。每个人经过一条边需要一定时间，每个节点提供的椅子有数量上限 C_i，每个节点提供两把椅子的时间至少相差 P_i。问所有人拿一把椅子回到根节点至少需要多少时间。

本题具有二分性

J: Join TB's Party

Problem Author: RogerDTZ

问题: 给定一棵边带权树，有 M 人从根节点出发，去各节点拿一把椅子回到根节点。每个人经过一条边需要一定时间，每个节点提供的椅子有数量上限 C_i，每个节点提供两把椅子的时间至少相差 P_i。问所有人拿一把椅子回到根节点至少需要多少时间。

本题具有二分性

预处理每个节点 i 到根的路径长度 len_i

J: Join TB's Party

Problem Author: RogerDTZ

问题: 给定一棵边带权树，有 M 人从根节点出发，去各节点拿一把椅子回到根节点。每个人经过一条边需要一定时间，每个节点提供的椅子有数量上限 C_i，每个节点提供两把椅子的时间至少相差 P_i。问所有人拿一把椅子回到根节点至少需要多少时间。

本题具有二分性

预处理每个节点 i 到根的路径长度 len_i

二分答案 T，若节点 i 满足 $2 * len_i \leq T$，则其能提供的椅子数量上限为

$$\min\{C_i, 1 + \lfloor \frac{T - 2 * len_i}{P_i} \rfloor\}$$

注意特判 $P_i = 0$ 的情况

J: Join TB's Party

Problem Author: RogerDTZ

问题: 给定一棵边带权树，有 M 人从根节点出发，去各节点拿一把椅子回到根节点。每个人经过一条边需要一定时间，每个节点提供的椅子有数量上限 C_i，每个节点提供两把椅子的时间至少相差 P_i。问所有人拿一把椅子回到根节点至少需要多少时间。

本题具有二分性

预处理每个节点 i 到根的路径长度 len_i

二分答案 T，若节点 i 满足 $2 * len_i \leq T$，则其能提供的椅子数量上限为

$$\min\{C_i, 1 + \lfloor \frac{T - 2 * len_i}{P_i} \rfloor\}$$

注意特判 $P_i = 0$ 的情况

时间复杂度 $O(N \log W)$，其中 W 为答案时间数量级

K: Koishi in Wetland Park

Problem Author: Satori

问题: 给定一张无向图，每个时刻边权会由 (a, b) 变为 $(b, a \bmod b)$，如果 b 为 0 则不变，点有点权。会以 $\frac{p_i}{\sum p_k}$ 的概率选择 i 作为起点，每次会等概率选择，走到 n 结束。问期望经过的边的权值 a 的和。

有 q 次修改，每次修改可能修改点权或边权，修改点权的次数不会超过 100.

K: Koishi in Wetland Park

Problem Author: Satori

问题: 给定一张无向图，每个时刻边权会由 (a, b) 变为 $(b, a \bmod b)$，如果 b 为 0 则不变，点有点权。会以 $\frac{p_i}{\sum p_k}$ 的概率选择 i 作为起点，每次会等概率选择，走到 n 结束。问期望经过的边的权值 a 的和。

有 q 次修改，每次修改可能修改点权或边权，修改点权的次数不会超过 100.

首先发现边权发生变化最多 log 次，可以单独处理完前 log 时间内的边权（阶段一），之后看作是一个不会变化的图上的随机游走（阶段二）。

K: Koishi in Wetland Park

Problem Author: Satori

问题: 给定一张无向图，每个时刻边权会由 (a, b) 变为 $(b, a \bmod b)$，如果 b 为 0 则不变，点有点权。会以 $\frac{p_i}{\sum p_k}$ 的概率选择 i 作为起点，每次会等概率选择，走到 n 结束。问期望经过的边的权值 a 的和。

有 q 次修改，每次修改可能修改点权或边权，修改点权的次数不会超过 100.

首先发现边权发生变化最多 log 次，可以单独处理完前 log 时间内的边权（阶段一），之后看作是一个不会变化的图上的随机游走（阶段二）。

这个时候考虑可以对于每条边计算其对答案的贡献。令 $p[x]$ 表示阶段二中点 x 期望被走过的次数。对于边 u, v, a, b 那么阶段二产生的贡献即为 $[u \neq n]\frac{p_u}{deg_u} + [v \neq n]\frac{p_v}{deg_v}$，对于阶段一可以用类似的方法计算边产生的贡献。

K: Koishi in Wetland Park

Problem Author: Satori

问题: 给定一张无向图，每个时刻边权会由 (a, b) 变为 $(b, a \bmod b)$，如果 b 为 0 则不变，点有点权。会以 $\frac{p_i}{\sum p_k}$ 的概率选择 i 作为起点，每次会等概率选择，走到 n 结束。问期望经过的边的权值 a 的和。

有 q 次修改，每次修改可能修改点权或边权，修改点权的次数不会超过 100.

首先发现边权发生变化最多 log 次，可以单独处理完前 log 时间内的边权（阶段一），之后看作是一个不会变化的图上的随机游走（阶段二）。

这个时候考虑可以对于每条边计算其对答案的贡献。令 $p[x]$ 表示阶段二中点 x 期望被走过的次数。对于边 u, v, a, b 那么阶段二产生的贡献即为 $[u \neq n]\frac{p_u}{deg_u} + [v \neq n]\frac{p_v}{deg_v}$，对于阶段一可以用类似的方法计算边产生的贡献。

p 要怎么求？

K: Koishi in Wetland Park

Problem Author: Satori

问题: 给定一张无向图，每个时刻边权会由 (a, b) 变为 $(b, a \bmod b)$，如果 b 为 0 则不变，点有点权。会以 $\frac{p_i}{\sum p_k}$ 的概率选择 i 作为起点，每次会等概率选择，走到 n 结束。问期望经过的边的权值 a 的和。

有 q 次修改，每次修改可能修改点权或边权，修改点权的次数不会超过 100.

首先发现边权发生变化最多 log 次，可以单独处理完前 log 时间内的边权（阶段一），之后看作是一个不会变化的图上的随机游走（阶段二）。

这个时候考虑可以对于每条边计算其对答案的贡献。令 $p[x]$ 表示阶段二中点 x 期望被走过的次数。对于边 u, v, a, b 那么阶段二产生的贡献即为 $[u \neq n]\frac{p_u}{deg_u} + [v \neq n]\frac{p_v}{deg_v}$，对于阶段一可以用类似的方法计算边产生的贡献。

p 要怎么求？

$p_x = \sum \frac{p_{to_x}}{deg_{to_x}} + st_x$ 其中 st_x 代表阶段一结束时，点在 x 的概率。

K: Koishi in Wetland Park

Problem Author: Satori

问题: 给定一张无向图，每个时刻边权会由 (a, b) 变为 $(b, a \bmod b)$，如果 b 为 0 则不变，点有点权。会以 $\frac{p_i}{\sum p_k}$ 的概率选择 i 作为起点，每次会等概率选择，走到 n 结束。问期望经过的边的权值 a 的和。

有 q 次修改，每次修改可能修改点权或边权，修改点权的次数不会超过 100.

首先发现边权发生变化最多 log 次，可以单独处理完前 log 时间内的边权（阶段一），之后看作是一个不会变化的图上的随机游走（阶段二）。

这个时候考虑可以对于每条边计算其对答案的贡献。令 $p[x]$ 表示阶段二中点 x 期望被走过的次数。对于边 u, v, a, b 那么阶段二产生的贡献即为 $[u \neq n]\frac{p_u}{deg_u} + [v \neq n]\frac{p_v}{deg_v}$，对于阶段一可以用类似的方法计算边产生的贡献。

p 要怎么求？

$p_x = \sum \frac{p_{to_x}}{deg_{to_x}} + st_x$ 其中 st_x 代表阶段一结束时，点在 x 的概率。

高斯消元即可

K: Koishi in Wetland Park

Problem Author: Satori

处理修改操作：

K: Koishi in Wetland Park

Problem Author: Satori

处理修改操作：

对于边权的修改：因为统计答案的时候是采用计算边的贡献获得，可以减去原有的边权的贡献 + 新的边权的贡献得到答案。

复杂度:O(log A) （对于阶段一需要枚举时刻，阶段二直接计算，A 代表边权的最大值）

K: Koishi in Wetland Park

Problem Author: Satori

处理修改操作：

对于边权的修改：因为统计答案的时候是采用计算边的贡献获得，可以减去原有的边权的贡献 + 新的边权的贡献得到答案。

复杂度:O(log A) （对于阶段一需要枚举时刻，阶段二直接计算，A 代表边权的最大值）

对于点权的修改：因为高斯消元部分的系数矩阵不发生变化，可以记录下高斯消元操作的过程，使得之后每次修改点权都只多计算 $O(n^2)$ 次

复杂度:$O(n^2 + logA \cdot m)$

K: Koishi in Wetland Park

Problem Author: Satori

处理修改操作：

对于边权的修改：因为统计答案的时候是采用计算边的贡献获得，可以减去原有的边权的贡献 + 新的边权的贡献得到答案。

复杂度:O(log A) （对于阶段一需要枚举时刻，阶段二直接计算，A 代表边权的最大值）

对于点权的修改：因为高斯消元部分的系数矩阵不发生变化，可以记录下高斯消元操作的过程，使得之后每次修改点权都只多计算 $O(n^2)$ 次

复杂度:$O(n^2 + logA \cdot m)$

总复杂度：$O(qlogA + n^3 + nm \cdot logA)$ (操作一次数和 n 同规模)

代码长度：3472

用时：

L: Look, BABA IS YOU

Problem Author: TmofZD

- **题目:** 在推箱子的规则上，额外增加一些文本块。若主谓宾紧挨着连成一条线，则该规则生效。要求按照指定指令序列模拟游戏过程。

L: Look, BABA IS YOU

Problem Author: TmofZD

- **题目:** 在推箱子的规则上，额外增加一些文本块。若主谓宾紧挨着连成一条线，则该规则生效。要求按照指定指令序列模拟游戏过程。
- **答案:** 模拟即可。

L: Look, BABA IS YOU

Problem Author: TmofZD

- **题目:** 在推箱子的规则上，额外增加一些文本块。若主谓宾紧挨着连成一条线，则该规则生效。要求按照指定指令序列模拟游戏过程。
- **答案:** 模拟即可。
- **额外信息:** Idea 来源自益智游戏"BABA IS YOU" 中的部分游玩方式。