正睿 OI 模拟赛题解 + 扩展

SRwudi

January 12, 2019

优美的闪电

题目大意

有 n 个带权值的区间。

你每次可以花费 c 的能量在任一坐标 x 出发射一枚导弹，可以将区间包含坐标 x，且权值都 $\leq c$ 的区间都消除。

询问最少话费多少能量能将所有区间都消除。

优美的闪电

题目大意

有 n 个带权值的区间。

你每次可以花费 c 的能量在任一坐标 x 出发射一枚导弹，可以将区间包含坐标 x，且权值都 $\leq c$ 的区间都消除。

询问最少话费多少能量能将所有区间都消除。

优美的闪电

$n \leq 20$

我们可以发现当前剩余权值最大的那个区间一定要被一发能量为该区间权值的导弹打掉，所以我们枚举该导弹击打位置，同时我们删除所有能被该导弹击打掉的区间。

优美的闪电

$n \leq 20$

我们可以发现当前剩余权值最大的那个区间一定要被一发能量为该区间权值的导弹打掉，所以我们枚举该导弹击打位置，同时我们删除所有能被该导弹击打掉的区间。

然后进行递归对剩余区间进行搜索。最后对搜索加上记忆化，用 $2^{2}0$ 的状态表示当前剩余导弹被消灭的最小能量值。

优美的闪电

$n \leq 20$

我们可以发现当前剩余权值最大的那个区间一定要被一发能量为该区间权值的导弹打掉，所以我们枚举该导弹击打位置，同时我们删除所有能被该导弹击打掉的区间。

然后进行递归对剩余区间进行搜索。最后对搜索加上记忆化，用 2^20 的状态表示当前剩余导弹被消灭的最小能量值。

本质上就是个状态压缩 dp，用 2^20 的状态表示当前剩余导弹被消灭的最小能量值，然后每次枚举能量为最大区间的导弹击打位置。

优美的闪电

$n \leq 20$

我们可以发现当前剩余权值最大的那个区间一定要被一发能量为该区间权值的导弹打掉，所以我们枚举该导弹击打位置，同时我们删除所有能被该导弹击打掉的区间。

然后进行递归对剩余区间进行搜索。最后对搜索加上记忆化，用 2^20 的状态表示当前剩余导弹被消灭的最小能量值。

本质上就是个状态压缩 dp，用 2^20 的状态表示当前剩余导弹被消灭的最小能量值，然后每次枚举能量为最大区间的导弹击打位置。

时间复杂度 $O(n \times 2^n)$。30 分

优美的闪电

测试点 6、7

两个测试点满足任意两个区间不相交。

优美的闪电

测试点 6、7

两个测试点满足任意两个区间不相交。
于是区间的包含关系变成父子关系，形成了若干棵树。

优美的闪电

测试点 6、7

两个测试点满足任意两个区间不相交。
于是区间的包含关系变成父子关系，形成了若干棵树。
来个例子？

优美的闪电

测试点 6、7

两个测试点满足任意两个区间不相交。
于是区间的包含关系变成父子关系，形成了若干棵树。
来个例子？
结论 1：总共有且仅有叶子数目枚导弹发射。

优美的闪电

测试点 6、7

两个测试点满足任意两个区间不相交。
于是区间的包含关系变成父子关系，形成了若干棵树。
来个例子？
结论 1：总共有且仅有叶子数目枚导弹发射。
剩下的则需要去分配每枚导弹能量的大小（树形 dp）。

优美的闪电

测试点 6、7

两个测试点满足任意两个区间不相交。

于是区间的包含关系变成父子关系，形成了若干棵树。

来个例子？

结论 1：总共有且仅有叶子数目枚导弹发射。

剩下的则需要去分配每枚导弹能量的大小（树形 dp）。

对于权值为 w 的节点（区间），考虑该节点所有子树，如果发射所有叶子节点导弹的能量为 $a_1 > a_2 > a_3 > \ldots > a_k$. 如果该节点权值 $\leq a_1$，表示发射给叶子的导弹能消除该父亲节点，不用管。如果该节点的权值 $> a_1$，则表示发射给叶子的所有导弹都无法消除该节点，所以最优操作是将 a_1 改成 w，这样总和最小。

优美的闪电

测试点 6、7

两个测试点满足任意两个区间不相交。

于是区间的包含关系变成父子关系，形成了若干棵树。

来个例子？

结论 1：总共有且仅有叶子数目枚导弹发射。

剩下的则需要去分配每枚导弹能量的大小（树形 dp）。

对于权值为 w 的节点（区间），考虑该节点所有子树，如果发射所有叶子节点导弹的能量为 $a_1 > a_2 > a_3 > ... > a_k$. 如果该节点权值 $\leq a_1$，表示发射给叶子的导弹能消除该父亲节点，不用管。如果该节点的权值 $> a_1$，则表示发射给叶子的所有导弹都无法消除该节点，所以最优操作是将 a_1 改成 w，这样总和最小。

则在 dp 的时候记录以一个节点为根的子树中，最少总能量，以及最少总能量中最大导弹的能量大小（尽可能大）。

优美的闪电

满分做法

使用区间 dp，$f_{i,j}$ 表示坐标被 $[i,j]$ **完全包含** 的区间最小需要花费的能量。

优美的闪电

满分做法

使用区间 dp，$f_{i,j}$ 表示坐标被 $[i,j]$ **完全包含** 的区间最小需要花费的能量。

则可以用暴力时候的思想，最大权值的区间需要一发单独的，且能量为该权值的导弹去消除。

优美的闪电

满分做法

使用区间 dp, $f_{i,j}$ 表示坐标被 $[i,j]$ **完全包含** 的区间最小需要花费的能量。

则可以用暴力时候的思想，最大权值的区间需要一发单独的，且能量为该权值的导弹去消除。

则枚举该导弹发射的坐标 x, 则子问题变成了 $f_{i,x-1}$，$f_{x+1,j}$。因为跨过 x 的区间一定都会被打掉，因为我们使用的是最大权值的能量。剩余的区间被 $[i,x-1],[x+1,j]$ 完全包含。

优美的闪电

满分做法

使用区间 dp，$f_{i,j}$ 表示坐标被 $[i,j]$ **完全包含** 的区间最小需要花费的能量。

则可以用暴力时候的思想，最大权值的区间需要一发单独的，且能量为该权值的导弹去消除。

则枚举该导弹发射的坐标 x，则子问题变成了 $f_{i,x-1}$，$f_{x+1,j}$。因为跨过 x 的区间一定都会被打掉，因为我们使用的是最大权值的能量。剩余的区间被 $[i,x-1],[x+1,j]$ 完全包含。

转移方程：

$$f_{i,j} = \min_{a_{Max_{i,j}} \le x \le b_{max_{i,j}}} \{f_{i,x-1} + f_{x+1,j} + w_{Max_{i,j}}\}$$

优美的闪电

满分做法

使用区间 dp，$f_{i,j}$ 表示坐标被 $[i,j]$ **完全包含** 的区间最小需要花费的能量。

则可以用暴力时候的思想，最大权值的区间需要一发单独的，且能量为该权值的导弹去消除。

则枚举该导弹发射的坐标 x，则子问题变成了 $f_{i,x-1}$，$f_{x+1,j}$。因为跨过 x 的区间一定都会被打掉，因为我们使用的是最大权值的能量。剩余的区间被 $[i,x-1],[x+1,j]$ 完全包含。

转移方程：

$$f_{i,j} = \min_{a_{Max_{i,j}} \le x \le b_{max_{i,j}}} \{f_{i,x-1} + f_{x+1,j} + w_{Max_{i,j}}\}$$

时间复杂度 $O(n^3)$。不过需要一点预处理：离散化坐标。预处理出被 $[i,j]$ 完全包含的最大权值的区间，即 $Max_{i,j}$。

优美的树

题目大意

给你一棵有边权的有根树。

每次你可以花费一点代价使某条边权 +1 或 -1。

问最少要花费多少代价使得从根节点出发到任意一个节点路径上的权值单调不下降。

$1 \le n \le 200000, 1 \le e_{x,y} \le 10^9$

优美的树

题目大意

给你一棵有边权的有根树。
每次你可以花费一点代价使某条边权 +1 或 -1。
问最少要花费多少代价使得从根节点出发到任意一个节点路径上的权值单调不下降。
$1 \le n \le 200000, 1 \le e_{x,y} \le 10^9$

欢迎吐槽。

暴力 dp

$O(n \times e_{x,y})$ 30 分。

暴力 dp

$O(n \times e_{x,y})$ 30 分。

智慧的结论

设开始所有边权的权值的集合为 S，那必然存在一种最优解使得调整过后的所有边的边权也在 S 集合中。

$O(n^2)$ 50 分。

尝试用函数去表示状态

不妨尝试用函数表示状态。

尝试用函数去表示状态

不妨尝试用函数表示状态。

a_i 表示 i 号节点到 i 号父亲的边的权值。

$f_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，末端权值均不小于 j 的最小代价。

$g_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，且 i 节点到 i 号父亲的边取值为 j 的最小代价。

$h_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，且 i 节点到 i 号父亲的边取值不小于 j 的最小代价。

尝试用函数去表示状态

不妨尝试用函数表示状态。

a_i 表示 i 号节点到 i 号父亲的边的权值。

$f_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，末端权值均不小于 j 的最小代价。

$g_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，且 i 节点到 i 号父亲的边取值为 j 的最小代价。

$h_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，且 i 节点到 i 号父亲的边取值不小于 j 的最小代价。

转移方程

$$f_{i,j} = \sum h_{son_i,j}$$

$$g_{i,j} = f_{i,j} + |a_i - j|$$

$$h_{i,j} = min(h_{i,j+1}, g_{i,j})$$

尝试用函数去表示状态

智慧的发现

- 函数均为下凸壳。
- 斜率均为整数。
- 需要支持操作叠加两个函数。(斜率对应增加)
- 以及求后缀最小值。(将负斜率清 0)

尝试用函数去表示状态

智慧的发现

- 函数均为下凸壳。
- 斜率均为整数。
- 需要支持操作叠加两个函数。(斜率对应增加)
- 以及求后缀最小值。(将负斜率清 0)

斜率为整数且单调从左向右递增，不妨维护拐点以及纵截距即可。

所以只需要支持拐点合并以及将负斜率部分的拐点删除两个操作即可。

尝试用函数去表示状态

智慧的发现

- 函数均为下凸壳。
- 斜率均为整数。
- 需要支持操作叠加两个函数。(斜率对应增加)
- 以及求后缀最小值。(将负斜率清 0)

斜率为整数且单调从左向右递增，不妨维护拐点以及纵截距即可。

所以只需要支持拐点合并以及将负斜率部分的拐点删除两个操作即可。

启发式平衡树？(low 爆了)

尝试用函数去表示状态

智慧的发现

- 函数均为下凸壳。
- 斜率均为整数。
- 需要支持操作叠加两个函数。(斜率对应增加)
- 以及求后缀最小值。(将负斜率清 0)

斜率为整数且单调从左向右递增，不妨维护拐点以及纵截距即可。

所以只需要支持拐点合并以及将负斜率部分的拐点删除两个操作即可。

启发式平衡树？(low 爆了)

小根可并堆（优秀）

尝试用函数去表示状态

智慧的发现

- 函数均为下凸壳。
- 斜率均为整数。
- 需要支持操作叠加两个函数。(斜率对应增加)
- 以及求后缀最小值。(将负斜率清 0)

斜率为整数且单调从左向右递增，不妨维护拐点以及纵截距即可。

所以只需要支持拐点合并以及将负斜率部分的拐点删除两个操作即可。

启发式平衡树？(low 爆了)

小根可并堆（优秀）

什么？你会 pb_ds？(强到爆炸！)

例题

APIO2016 Fireworks

给你一棵有边权的有根树。

每次你可以花费一点代价使某条边权 +1 或 -1。

问最少要花费多少代价使得从根节点出发到任意一个节点路径上的权值均相同。

$1 \leq n \leq 200000, 1 \leq e_{x,y} \leq 10^9$

例题

APIO2016 Fireworks

给你一棵有边权的有根树。

每次你可以花费一点代价使某条边权 +1 或 -1。

问最少要花费多少代价使得从根节点出发到任意一个节点路径上的权值均相同。

$1 \le n \le 200000, 1 \le e_{x,y} \le 10^9$

给大家 5min 思考时间。

依然用函数去表示状态

表示状态。

$f_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，末端权值均不小于 j 的最小代价。

依然用函数去表示状态

表示状态。

$f_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，末端权值均不小于 j 的最小代价。

我们需要支持操作叠加两个凸壳，以及将一个凸壳延另外一个凸壳轨迹延展后求并。

观察这个下凸壳的特殊性质。

依然用函数去表示状态

表示状态。

$f_{i,j}$ 代表到了以 i 为根的子树已经处理完毕，末端权值均不小于 j 的最小代价。

我们需要支持操作叠加两个凸壳，以及将一个凸壳延另外一个凸壳轨迹延展后求并。

观察这个下凸壳的特殊性质。

维护大根可并堆。

每次弹出最右端的两个拐点，再重新插入两个新的拐点即可完成延展操作。

hdu5343 MZL's Circle Zhou

题目大意

给你两个母串 A, B。

分别截取 A 的子串 X 和 B 的子串 Y，连接组成 $X+Y$，求本质不同的 $X+Y$ 的方案数。(子串可以为空)

$|A|, |B| \leq 10^5$

hdu5343 MZL's Circle Zhou

题目大意

给你两个母串 A, B。

分别截取 A 的子串 X 和 B 的子串 Y，连接组成 $X+Y$，求本质不同的 $X+Y$ 的方案数。(子串可以为空)

$|A|, |B| \leq 10^5$

给大家 5min 思考时间。

hdu5343 MZL's Circle Zhou

对于一个合法的串，其合法的分割方式必然是一段连续的区间，所以我们每次在分割方式最靠后的位置上计数即可。

hdu5343 MZL's Circle Zhou

对于一个合法的串，其合法的分割方式必然是一段连续的区间，所以我们每次在分割方式最靠后的位置上计数即可。

直接在子串的自动机上面套 dp 即可。

hdu5343 MZL's Circle Zhou

对于一个合法的串，其合法的分割方式必然是一段连续的区间，所以我们每次在分割方式最靠后的位置上计数即可。

直接在子串的自动机上面套 dp 即可。

因此，对两个字符串分别建立子串的自动机 A 和 B，对于 A 中找子串 X，当 X 的末尾不能接某个字符 c 时，那便转移到 B 中以 c 为开头的子串。

hdu5343 MZL's Circle Zhou

对于一个合法的串，其合法的分割方式必然是一段连续的区间，所以我们每次在分割方式最靠后的位置上计数即可。

直接在子串的自动机上面套 dp 即可。

因此，对两个字符串分别建立子串的自动机 A 和 B，对于 A 中找子串 X，当 X 的末尾不能接某个字符 c 时，那便转移到 B 中以 c 为开头的子串。

暴力构建子串的字典树 $O(n^2)$

hdu5343 MZL's Circle Zhou

对于一个合法的串，其合法的分割方式必然是一段连续的区间，所以我们每次在分割方式最靠后的位置上计数即可。

直接在子串的自动机上面套 dp 即可。

因此，对两个字符串分别建立子串的自动机 A 和 B，对于 A 中找子串 X，当 X 的末尾不能接某个字符 c 时，那便转移到 B 中以 c 为开头的子串。

暴力构建子串的字典树 $O(n^2)$

后缀自动机 $O(n \times |Char|)$

优美的串 By：SRwudi

题目大意

给你两个母串 A 和 B。

若一个串存在一种分割方式，使得每一部分均为 A 的子串或 B 的子串，且排列的顺序为 A 的子串和 B 的子串交替排列，则称作这个串为优美的串。

即排列方式如下：

$$A[l_1, r_1]B[l_2, r_2]A[l_3, r_3]B[l_4, r_4].......A[l_{k-1}, r_{k-1}]B[l_k, r_k]$$

子串不能为空。且必须以 A 的子串为开头，以 B 的子串为结尾。

问长度不超过 K 的优美的串的个数。

优美的串 By：SRwudi

题目大意

给你两个母串 A 和 B。

若一个串存在一种分割方式，使得每一部分均为 A 的子串或 B 的子串，且排列的顺序为 A 的子串和 B 的子串交替排列，则称作这个串为优美的串。

即排列方式如下：

$$A[l_1, r_1]B[l_2, r_2]A[l_3, r_3]B[l_4, r_4]......A[l_{k-1}, r_{k-1}]B[l_k, r_k]$$

子串不能为空。且必须以 A 的子串为开头，以 B 的子串为结尾。

问长度不超过 K 的优美的串的个数。

欢迎吐槽。

部分分大讨论

测试点 1

输出 $\lfloor \frac{K}{2} \rfloor$。期望得分 5 分。

部分分大讨论

测试点 1

输出 $\lfloor \frac{K}{2} \rfloor$。期望得分 5 分。

测试点 2

$O(4^K \times n^4)$ 枚举 + 动归。结合测试点 1 期望得分 20 分。

部分分大讨论

测试点 1

输出 $\lfloor \frac{K}{2} \rfloor$。期望得分 5 分。

测试点 2

$O(4^K \times n^4)$ 枚举 + 动归。结合测试点 1 期望得分 20 分。

怎么暴力只有 20 分，什么辣鸡出题人！

各自包含独立的字符如何设计状态

我们发现对于一个合法串的分割方式是固定的。

各自包含独立的字符如何设计状态

我们发现对于一个合法串的分割方式是固定的。

母串 A 只包含字母 A，母串 B 只包含字母 T

设计状态：

$f_{i,S}$ 表示长度为 i，按照 $ABAB...$ 方式排列结尾情况为 S 的串的个数。S 记录当前串的结尾有连续多少个 A 或连续个多少个 T。

那 S 的状态数一共为 $2n$ 个。

对于 $K \leq 500$ 直接 dp，对于 $K \leq 10^9$ 矩乘 + 前缀和。

各自包含独立的字符如何设计状态

母串 A 只包含字母 A, C，母串 B 只包含字母 G, T

设计状态：

$f_{i,S}$ 表示长度为 i，按照 $ABAB...$ 方式排列结尾情况为 S 的串的个数。S 记录当前串的结尾是 A 中某个子串，还是 T 中某个子串。

那 S 的状态数一共为 n^2 个，通过后缀自动机优化一共为 $2n$ 个。

对于 $K \leq 500$ 直接 dp，对于 $K \leq 10^9$ 矩乘 + 前缀和。

各自包含独立的字符如何设计状态

母串 A 只包含字母 A, C，母串 B 只包含字母 G, T

设计状态：

$f_{i,S}$ 表示长度为 i，按照 $ABAB...$ 方式排列结尾情况为 S 的串的个数。S 记录当前串的结尾是 A 中某个子串，还是 T 中某个子串。

那 S 的状态数一共为 n^2 个，通过后缀自动机优化一共为 $2n$ 个。

对于 $K \leq 500$ 直接 dp，对于 $K \leq 10^9$ 矩乘 + 前缀和。

暴力就有 60 分，哪里去找这么好的良心出题人！

共享同样的字符如何设计状态

对于有共享字符的如何建立状态？

共享同样的字符如何设计状态

对于有共享字符的如何建立状态？

建立状态的难点

- 发现对于一个串可以有多种 $ABABAB....$ 的排列方式。
- 一个串可以分割成以 A 子串结尾的 $ABAB...A$ 的排列，也可以分割成以 B 子串结尾的 $ABAB...AB$ 的排列。
- 在以 A 子串结尾的 $ABAB...A$ 的分割模式下，可能结尾 A 子串不同。B 也同理。

共享同样的字符如何设计状态

共享同样的字符如何设计状态

那如此错综复杂的状态，将如何记录。

共享同样的字符如何设计状态

那如此错综复杂的状态，将如何记录。

建立状态

f_{i,S_A,S_B} 记录长度为 i 的，所有以结尾为 A 子串的分割方式中最短的子串状态为 S_A（若不存在以 A 结尾的分割方式，则 S_A 为空），所有以结尾为 B 子串的分割方式中最短的子串状态为 S_B（若不存在以 B 结尾的分割方式，则 S_B 为空）的串的方案数。

共享同样的字符如何设计状态

那如此错综复杂的状态，将如何记录。

建立状态

f_{i,S_A,S_B} 记录长度为 i 的，所有以结尾为 A 子串的分割方式中最短的子串状态为 S_A（若不存在以 A 结尾的分割方式，则 S_A 为空），所有以结尾为 B 子串的分割方式中最短的子串状态为 S_B （若不存在以 B 结尾的分割方式，则 S_B 为空）的串的方案数。

思考：为什么记录最短子串的状态即可？

共享同样的字符如何设计状态

那如此错综复杂的状态，将如何记录。

建立状态

f_{i,S_A,S_B} 记录长度为 i 的，所有以结尾为 A 子串的分割方式中最短的子串状态为 S_A（若不存在以 A 结尾的分割方式，则 S_A 为空），所有以结尾为 B 子串的分割方式中最短的子串状态为 S_B （若不存在以 B 结尾的分割方式，则 S_B 为空）的串的方案数。

思考：为什么记录最短子串的状态即可？
子串的延展性。

共享同样的字符如何设计状态

那如此错综复杂的状态，将如何记录。

建立状态

f_{i,S_A,S_B} 记录长度为 i 的，所有以结尾为 A 子串的分割方式中最短的子串状态为 S_A（若不存在以 A 结尾的分割方式，则 S_A 为空），所有以结尾为 B 子串的分割方式中最短的子串状态为 S_B （若不存在以 B 结尾的分割方式，则 S_B 为空）的串的方案数。

思考：为什么记录最短子串的状态即可？

子串的延展性。

用后缀自动机表示子串的状态，状态总数为 $O(4 \times n^2)$，对于 $K \leq 500$ 直接 dp，对于 $K \leq 10^9$ 矩乘 + 前缀和。可以通过 60 分。

共享同样的字符如何设计状态

那如此错综复杂的状态，将如何记录。

建立状态

f_{i,S_A,S_B} 记录长度为 i 的，所有以结尾为 A 子串的分割方式中最短的子串状态为 S_A（若不存在以 A 结尾的分割方式，则 S_A 为空），所有以结尾为 B 子串的分割方式中最短的子串状态为 S_B（若不存在以 B 结尾的分割方式，则 S_B 为空）的串的方案数。

思考：为什什么记录最短子串的状态即可？

子串的延展性。

用后缀自动机表示子串的状态，状态总数为 $O(4 \times n^2)$，对于 $K \leq 500$ 直接 dp，对于 $K \leq 10^9$ 矩乘 + 前缀和。可以通过 60 分。

搞了半天怎么还只有 60 分？什么蒟蒻出题人！

压缩状态

考虑转移：我们设当前子串的状态为 (S_A, S_B)，才后面添加 A, C, G, T 字符进行转移。

压缩状态

考虑转移：我们设当前子串的状态为 (S_A, S_B)，才后面添加 A, C, G, T 字符进行转移。

设 $trans_{S,ch}$ 表示子串状态 S 经过 ch 转移到的节点。

- 若当前转移的字符 A, B 均不包含，则 (S_A, S_B) 转移到 $(null\ null)$。
- 若当前转移的字符 A 包含，B 不包含，若 S_B 不为空则 (S_A, S_B) 转移到 $(trans_{root_A,ch}, null)$，否则转移到 $(trans_{S_A,ch}, null)$。当前转移的字符 B 包含，A 不包含同理。
- 当前转移的字符 A, B 均包含，若 S_A, S_B 均不为空则转移到 $(trans_{root_A,ch}, trans_{root_B,ch})$，若 S_B 为空，则转移到 $(trans_{S_A,ch}, trans_{root_B,ch})$。$S_A$ 为空同理。

压缩状态

考虑转移：我们设当前子串的状态为 (S_A, S_B)，才后面添加 A, C, G, T 字符进行转移。

设 $trans_{S,ch}$ 表示子串状态 S 经过 ch 转移到的节点。

- 若当前转移的字符 A, B 均不包含，则 (S_A, S_B) 转移到 $(null\ null)$。
- 若当前转移的字符 A 包含，B 不包含，若 S_B 不为空则 (S_A, S_B) 转移到 $(trans_{root_A,ch}, null)$，否则转移到 $(trans_{S_A,ch}, null)$。当前转移的字符 B 包含，A 不包含同理。
- 当前转移的字符 A, B 均包含，若 S_A, S_B 均不为空则转移到 $(trans_{root_A,ch}, trans_{root_B,ch})$，若 S_B 为空，则转移到 $(trans_{S_A,ch}, trans_{root_B,ch})$。$S_A$ 为空同理。

那我们发现可以到达的状态满足 S_A, S_B 其中一个为空或者 S_A, S_B 为某个初始点经过一个字符转移边到达的状态。

压缩状态

考虑转移：我们设当前子串的状态为 (S_A, S_B)，才后面添加 A, C, G, T 字符进行转移。

设 $trans_{S,ch}$ 表示子串状态 S 经过 ch 转移到的节点。

- 若当前转移的字符 A, B 均不包含，则 (S_A, S_B) 转移到 $(null\ null)$。
- 若当前转移的字符 A 包含，B 不包含，若 S_B 不为空则 (S_A, S_B) 转移到 $(trans_{root_A,ch}, null)$，否则转移到 $(trans_{S_A,ch}, null)$。当前转移的字符 B 包含，A 不包含同理。
- 当前转移的字符 A, B 均包含，若 S_A, S_B 均不为空则转移到 $(trans_{root_A,ch}, trans_{root_B,ch})$，若 S_B 为空，则转移到 $(trans_{S_A,ch}, trans_{root_B,ch})$。$S_A$ 为空同理。

那我们发现可以到达的状态满足 S_A, S_B 其中一个为空或者 S_A, S_B 为某个初始点经过一个字符转移边到达的状态。

由于字符集为 4，所以总状态数削减到了 $5 \times 2 \times 2n$。期望得分 100 分。

bzoj4180 字符串计数

题目大意

给你一母串 T，T 中有且仅有 4 种字符 A, B, C, D 。

你可以通过构造一个新的字符串 S，构造的方法是：进行多次操作，每一次操作选择 T 的一个子串，将其加入 S 的末尾。

对于一个可构造出的字符串 S，可能有多种构造方案，定义构造字符串 S 所需的操作次数为所有构造方案中操作次数的最小值。

给定的正整数 N 和字符串 T，求能构造出的所有长度为 N 的字符串 S 中，所需的操作次数最大的字符串的操作次数。

bzoj4180 字符串计数

题目大意

给你一母串 T，T 中有且仅有 4 种字符 A, B, C, D 。

你可以通过构造一个新的字符串 S，构造的方法是：进行多次操作，每一次操作选择 T 的一个子串，将其加入 S 的末尾。

对于一个可构造出的字符串 S，可能有多种构造方案，定义构造字符串 S 所需的操作次数为所有构造方案中操作次数的最小值。

给定的正整数 N 和字符串 T，求能构造出的所有长度为 N 的字符串 S 中，所需的操作次数最大的字符串的操作次数。

给个样例思考一下 $T = AAABB, N = 10$，最大的构造次数？

bzoj4180 字符串计数

构造矩阵 $S_{i,j}$，表示第 i 个字母到第 j 个字母，最短的非 S 子串的长度。

bzoj4180 字符串计数

构造矩阵 $S_{i,j}$，表示第 i 个字母到第 j 个字母，最短的非 S 子串的长度。

定义“矩阵乘法”：$C_{i,j} = \min\left\{A_{i,k} + B_{k,j}\right\}$

此乘法满足结合律，故可以用快速幂加速运算。

bzoj4180 字符串计数

构造矩阵 $S_{i,j}$，表示第 i 个字母到第 j 个字母，最短的非 S 子串的长度。

定义"矩阵乘法"：$C_{i,j} = \min \left\{ A_{i,k} + B_{k,j} \right\}$

此乘法满足结合律，故可以用快速幂加速运算。

那么 $S^k_{i,j}$ 记录的是第 i 个字母到第 j 个字母操作数为 k 次能构造出来的长度最小的串。

bzoj4180 字符串计数

构造矩阵 $S_{i,j}$，表示第 i 个字母到第 j 个字母，最短的非 S 子串的长度。

定义“矩阵乘法”：$C_{i,j} = \min\left\{A_{i,k} + B_{k,j}\right\}$

此乘法满足结合律，故可以用快速幂加速运算。

那么 $S^k_{i,j}$ 记录的是第 i 个字母到第 j 个字母操作数为 k 次能构造出来的长度最小的串。

于是我们采用二分答案，二分一个操作数，然后用矩阵快速幂做操作数次，判断一下最小长度是否 $\geq n$ 即可。

bzoj4180 字符串计数

构造矩阵 $S_{i,j}$，表示第 i 个字母到第 j 个字母，最短的非 S 子串的长度。

定义“矩阵乘法”：$C_{i,j} = \min\left\{A_{i,k} + B_{k,j}\right\}$

此乘法满足结合律，故可以用快速幂加速运算。

那么 $S^k_{i,j}$ 记录的是第 i 个字母到第 j 个字母操作数为 k 次能构造出来的长度最小的串。

于是我们采用二分答案，二分一个操作数，然后用矩阵快速幂做操作数次，判断一下最小长度是否 $\geq n$ 即可。

矩阵 $S_{i,j}$ 怎么求？后缀自动机上的简单 dp。