Replace

给定由小写英文字母组成的字符串S，T和K种操作。最开始当前字符串是S，可以以任意顺序使用任意次K种操作来修改当前字符串，其中第i种操作可以把当前字符串中的一个**字符**C_i替换为**字符串**A_i。最少要几次操作可以把S变成T？如果无解输出-1。

$1 \le |S| \le |T| \le 50, K \le 50, 1 \le |A_i| \le 50$

```
ab
cbca
3
a b
b ca
a efg
```

ab->bb->bca->caca->cbca

Replace

给定由小写英文字母组成的字符串S，T和K种操作。最开始当前字符串是S，可以以任意顺序使用任意次K种操作来修改当前字符串，其中第i种操作可以把当前字符串中的一个**字符**C_i替换为**字符串**A_i。最少要几次操作可以把S变成T？如果无解输出-1。

$1 \le |S| \le |T| \le 50, K \le 50$，$1 \le |A_i| \le 50$

分析S到T的变换过程，整体来看，最终是S里的每个字符变成T中的一段区间。

于是设$dp_{i,j}$表示S的前i个字符对应T的前j个，转移枚举S_i对应的区间长度。

所以需要求$f_{ch,\,L,\,R}$表示字符ch变成$T[L,R]$的最少次数。

分析字符ch到$T[L,R]$的变换过程，某次操作从长度1变成长度大于1的字符串。

所以设计$g_{i,\,j,\,L,\,R}$表示字符串A_i长为j的前缀，变成$T[L,R]$的最少次数，转移枚举$A_{i,j}$对应的区间长度。

$$g_{i,\,j,\,L,\,R} = \min_{L \le k < R} \left(g_{i,\,j-1,\,L,\,k} + f_{A_{i,j},\,k+1,\,R} \right)$$

当前状态依赖$R-L+1$更小的状态，我们按$R-L+1$从小到大的顺序求。

实现时可以增加操作$\beth \to S$，不用做第一个dp。

效率$50^5/3$!

Cigar Box

一个数列一开始是$(1,2,\ldots,n)$，接下来进行m次操作，每次操作把数列中的一个数移动到数列开头或者末尾。有几种长度为m的操作序列能使最终数列变成$(a_1,a_2,\ldots,a_n)$?
两种操作序列被视为相同当且仅当每次操作选择了相同的数且移动目标相同。
$n,m \le 3000$，$a_1,a_2,\ldots,a_n$是1..n的排列。

Cigar Box

一个数列一开始是$(1,2,\dots,n)$，接下来进行m次操作，每次操作把数列中的一个数移动到数列开头或者末尾。有几种长度为m的操作序列能使最终数列变成$(a_1,a_2,\dots,a_n)$?
两种操作序列被视为相同当且仅当每次操作选择了相同的数且移动目标相同。
$n,m\le 3000$，$a_1,a_2,\dots,a_n$是1. . n的排列。

对某个数字x进行的操作中，只有最后一次操作是有效的。

我们只需要确定哪些位置的操作是同种数字，具体数字会根据有效操作在操作序列中的位置自动对应。

设有L个有效操作往左丢，R个有效操作往右丢，那么$a_{L+1}\dots a_{n-R}$要递增。

考虑从右往左逐位确定操作序列，$f_{i,L,R}$表示考虑完后i位，有L个有效操作往左丢，R个有效操作往右丢的方案数。枚举下一位是有效操作，还是跟后面某个有效操作同类的无效操作。

我们需要的只是$f_{m,\ 1..n,\ 1..n}$的这些状态

可以先不确定每个有效操作是往左丢还是往右丢，即$g_{i,j}$表示考虑完后i位有j个有效操作，最后算$f_{m,\ 1..n,\ 1..n}$这些状态时再确定每个有效操作是什么方向，也就是乘上一个组合数。

背包

有 M 个背包容量分别为 X_i 千克， N 个物品的重量分别为 Y_i 千克，请问最少需要多少个背包可以把 N 个物品全部装进去呢？如果装不下请输出-1。

$1 \le N \le 20, 1 \le M \le 50, 1 \le X_i, Y_i \le 10^8$

背包

有 M 个背包容量分别为 X_i 千克， N 个物品的重量分别为 Y_i 千克，请问最少需要多少个背包可以把 N 个物品全部装进去呢？如果装不下请输出-1。
$1 \le N \le 20, 1 \le M \le 50, 1 \le X_i, Y_i \le 10^8$

一定选择最大的一些背包，先把背包从大到小排序。

依次确定每个背包装的物品状态，$f_{i,S}$表示前i个包能否装状态为S的物品，转移枚举下个背包装哪些物品

转移的数量太多了，考虑每次只装一个物品。

那么每次从还没装入的物品里选一个，装当前背包或者装下一个新的空背包。

为了判断能否装当前背包，状态要记当前背包的剩余容量，$f_{i,S,last}$表示前i个背包装状态为S的物品，第i个背包剩下空间为last是否可行。

在$f_{i,S,*}$所有状态中，显然剩余空间最大的那个状态是最好的。所以只要求出$f_{i,S}$表示前i个背包装入状态为S的物品，第i个包的最多剩余空间。复杂度$O(n^2 2^n)$。

高楼建造

某地正在规划建造N座高楼，它们要被安排在一条笔直的街道上，且高度是1~N的一个排列.
街道上从左到右有1~X共X个可供选址的位置，相邻位置之间距离为1.
规划方案要满足对于任意高楼，左边高楼到它的距离和右边高楼到它的距离均大于等于它的高度. 请问有多少可能的规划方案?
$1 \le N \le 100,\ 1 \le X \le 10^5$

高楼建造

某地正在规划建造N座高楼，它们要被安排在一条笔直的街道上，且高度是1~N的一个排列.
街道上从左到右有1~X共X个可供选址的位置，相邻位置之间距离为1.
规划方案要满足对于任意高楼，左边高楼到它的距离和右边高楼到它的距离均大于等于它的高度. 请问有多少可能的规划方案?
$1 \le N \le 100,\ 1 \le X \le 10^5$

考虑先确定N座楼之间的顺序，再确定具体位置。
使用简单组合技巧可知，放具体位置的方案数只跟$\sum \max(h_i, h_{i+1})$有关。
考虑对N^2个$\sum \max(h_i, h_{i+1})$的取值，求对应的排列数量。
我们关心相邻的值，所以考虑从小到大放入数字的决策过程，状态记一维跟$\sum \max(h_i, h_{i+1})$ 相关的信息。
但是每次加入数字会使一个旧的间隔消失，而我们没有旧的间隔相关信息。
我们直接规定某些间隔不能消失，会保留到最后。状态中记下这些间隔的和。
插入一个数字时需要知道可以插入的位置数量，所以状态里记不能消失的间隔数量。
$f[i][j][k]$表示插入前i个数，有j个间隔保留到最后，这j个间隔的和是k的方案数。

菜园

菜园中从东到西有N株植物，第i株植物的高度是H_i。春天时可以花费C_i元拔去第i株植物。如果一株植物左边$(1 \ldots i-1)$没有比它高的植物，或者右边$(i+1 \ldots N)$没有比它高的植物，经过夏季阳光的充分照射，它会在秋天结果，可以获得P_i元收益。如何打理菜园能获得最大收益？

$3 \le N \le 10^5, 1 \le H_i, P_i, C_i \le 10^9$

菜园

菜园中从东到西有N株植物，第i株植物的高度是H_i。春天时可以花费C_i元拔去第i株植物。
如果一株植物左边$(1 \dots i-1)$没有比它高的植物，或者右边$(i+1 \dots N)$没有比它高的植物，经过夏季阳光的充分照射，它会在秋天结果，可以获得P_i元收益。如何打理菜园能获得最大收益？

$3 \le N \le 10^5, 1 \le H_i, P_i, C_i \le 10^9$

结果的植物高度一定先递增再递减。 于是考虑算每个位置结尾的递增序列的最大收益。递减同理。
dp_i表示以位置i结尾的递增序列的最大收益，从高度$\le H_i$的位置转移过来。
这让我们想到用个辅助数组维护前面的状态对后续的贡献。
g_{h}表示前面的状态对后续一个高度为h的位置，最大转移是多少。
那么$dp_i = g_{H_i} + P_i$，
经过第i个位置，所有$g[1..H_i - 1]$都要减去C_i，然后$g[H_i..N]$对dp_i取max。
这些操作都可以使用线段树维护。

排列

有两个排列$A_{1..n}$和$B_{1..n}$，这两个排列的分数为满足$A_i > B_i$的i的数量.
由于数据损坏，对于每一个i，A_i和B_i中的恰好一个数字丢失了，丢失的数字用0表示.
有几种恢复的方法满足A和B仍然是排列且分数为S?
$0 \leq S < n \leq 5000$

例如：
4 2
4 2 0 0
0 0 4 2
可以是4 2 1 3\n1 3 4 2，也可以是4 2 3 1\n3 1 4 2，所以答案是2。

排列

有两个排列$A_{1..n}$和$B_{1..n}$，这两个排列的分数为满足$A_i > B_i$的i的数量.
由于数据损坏，对于每一个i，A_i和B_i中的恰好一个数字丢失了，丢失的数字用0表示.
有几种恢复的方法满足A和B仍然是排列且分数为S?
$0 \le S < n \le 5000$

可以分成两个子问题，对于子问题，求出每种分数对应的方案数。

子问题可以表示为：给定数组A，B，任意匹配A和B，每种分数对应的匹配方案有多少？

分数恰好为某值的方案数不好求，考虑容斥。

$f_{i,j}$表示从小到大考虑前i个A，规定其中j个匹配了更小的B，其余暂不匹配的方案数。

$$f_{i,j} = f_{i-1,j} + f_{i-1,j-1} * (num_i - j + 1)$$

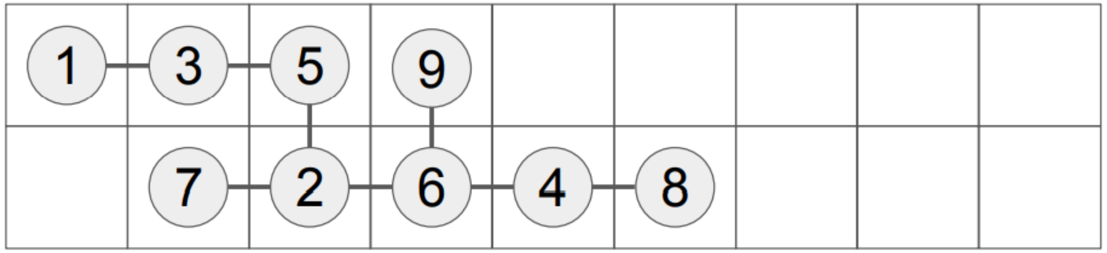
最终的$f_{n,j}$乘上$(n-j)!$，表示剩下的任意匹配。

此时$f_{n,j}$还不是分数恰好为j的方案数，设g_j表示分数恰好为j的方案数，

$$g_j = f_{n,j} - \sum_{k>j} g_k * C_k^j$$

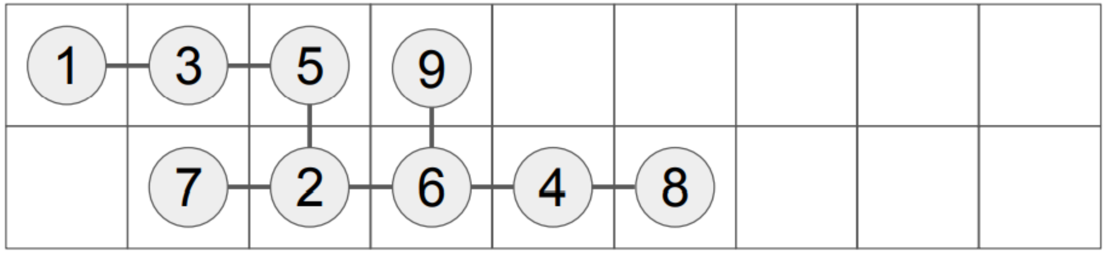
Embedding

给定一颗N个节点的树，把树嵌入$2 \times N$的网格中，使得节点1在左上角格子，且树上相邻的节点在网格中四相邻。有几种嵌入的方法？ $N \leq 3 \times 10^5$



Embedding

给定一颗N个节点的树，把树嵌入$2\times N$的网格中，使得节点1在左上角格子，且树上相邻的节点在网格中四相邻。有几种嵌入的方法？ $N\le 3\times 10^5$



dp_i表示以i为根的子树，把i放左上角对应的答案。

如果i有两个儿子，找到对应的两条链末端；

如果i有一个儿子j，放下面的转移显然；放右边是dp_j，补上缺少的一种情况。

Lights Out on Connected Graph

给定一个N个点M条边的简单连通无向图G。
在2^M种删边方式中，有几种能使剩下的图是连通的二分图？
$N \le 17, M \le N(N-1)/2$

Lights Out on Connected Graph

给定一个N个点M条边的简单连通无向图G。
在2^M种删边方式中，有几种能使剩下的图是连通的二分图？
$N \le 17, M \le N(N-1)/2$

考虑交换枚举顺序，先确定哪些点在左半边，哪些点在右半边，然后在这些点之间的边里选一些形成二分图。但是这样确定的图可能不连通。

考虑算连通图的套路，枚举包含最小编号的连通块T。

先算g_S表示把点集S黑白染色，在连接黑白的边里选一些，不要求连通的方案数。
令f_S表示把点集S黑白染色，在连接黑白的边里选一些，形成连通的方案数。

$$f_S = g_S - \sum f_T * g_{S\setminus T}$$

Sequence Growing Hard

给定 n, k, m , 问有多少个序列组 $(A_0, A_1, \ldots, A_n)$ 满足：序列 A_i 的元素个数为 i ; 所有元素都在 $[1, k]$ 内; $\forall i \in [0, n)$, A_i 是 A_{i+1} 的子序列且 A_i 的字典序小于 A_{i+1}.
输出在 $\bmod m$ 意义下的答案.

$1 \le N, K \le 300$
$2 \le M \le 10^9$

例如$N = K = 2$，有5种序列组：
(), (1), (1, 1)
(), (1), (1, 2)
(), (1), (2, 1)
(), (2), (2, 1)
(), (2), (2, 2)

Sequence Growing Hard

给定 n, k, m，问有多少个序列组 $(A_0, A_1, \ldots, A_n)$ 满足：序列 A_i 的元素个数为 i；所有元素都在 $[1, k]$ 内; $\forall i \in [0, n)$，A_i 是 A_{i+1} 的子序列且 A_i 的字典序小于 A_{i+1}.
输出在 $\bmod m$ 意义下的答案.

$1 \le N, K \le 300$
$2 \le M \le 10^9$

每次操作相当于在序列末尾或者一个$< x$的数前面插入x。

把x插到y前面可以理解为x挂到y下面，最终形成一颗树。

设$dp_{i,j}$表示有多少方法形成i个点，根节点数值为j的树。

枚举第一个子树的大小和根节点值进行转移。

病毒

某种病毒有一套独特的演化方式。设有编号为0到$G-1$共G种基因，一个病毒一开始只包含一个基因，发生变异时，当前基因序列中的一个基因被替换为一段基因序列。 0、1基因不会发生变异。研究员找出了所有可能的变异情况，称为变异规则，如下例：

$2 \rightarrow <0\ \ 1>$

$3 \rightarrow <2\ \ 0\ \ 0>$

$3 \rightarrow <1\ \ 4>$

$4 \rightarrow <0\ \ 3\ \ 1\ \ 2>$

病毒会不断变异直到基因序列中只含有0、1两种基因，称为稳定病毒。例如：

$<4> \rightarrow <\underline{0\ \ 3\ \ 1\ \ 2}> \rightarrow <0\ \ \underline{2\ \ 0\ \ 0}\ \ 1\ \ 2> \rightarrow <0\ \ \underline{0\ \ 1}\ \ 0\ \ 0\ \ 1\ \ 2> \rightarrow <0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 1\ \ \underline{0\ \ 1}>$

有M种抗体，第i种抗体对应01串s_i，所有包含s_i作为连续子串的稳定病毒都能被抗体i识别出来。

病毒的基因序列可能很长，甚至一直不成为稳定病毒。

给定变异规则和M种抗体，对于每个单独的基因，研究人员想知道所有从这个单独基因开始变异的，不能被M种抗体识别出来的稳定病毒中，最短的那个稳定病毒的长度是多少？

除了0、1，每个基因一定出现在某条变异规则左端；所有变异规则右端的长度之和不超过100，所有抗体长度之和不超过50。