概率和期望

huangkui

August 6, 2019

雅礼中学

写在前面

写在前面

- 水平有限，找了一些与真正概率期望相关的小套路来讲

写在前面

- 水平有限，找了一些与真正概率期望相关的小套路来讲
- 前面的一些基础知识有助于大家更深入地理解概率期望；后面的题目大多比较简单，可以放心食用

写在前面

- 水平有限，找了一些与真正概率期望相关的小套路来讲
- 前面的一些基础知识有助于大家更深入地理解概率期望；后面的题目大多比较简单，可以放心食用
- 大家放心，今天的讲课内容比大前天不知道简单到哪里去了，不过内容可能有点多，我会讲得比较慢，讲不完就霸占下午讲课的时间

基础知识

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件
- 理解：A 和 B **不能同时发生**, 即 $P(AB) = 0$

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件
- 理解：A 和 B **不能同时发生**, 即 $P(AB) = 0$
- 性质：若 A, B 互不相容, 则 $P(A \cup B) = P(A) + P(B)$

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件
- 理解：A 和 B **不能同时发生**, 即 $P(AB) = 0$
- 性质：若 A, B 互不相容, 则 $P(A \cup B) = P(A) + P(B)$
- **对立事件**：若 A, B 互斥, 且 $P(A) + P(B) = 1$, 则称 A 和 B 为对立事件

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件
- 理解：A 和 B **不能同时发生**, 即 $P(AB) = 0$
- 性质：若 A, B 互不相容, 则 $P(A \cup B) = P(A) + P(B)$
- **对立事件**：若 A, B 互斥, 且 $P(A) + P(B) = 1$, 则称 A 和 B 为对立事件
- 举个例子：

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件
- 理解：A 和 B **不能同时发生**, 即 $P(AB) = 0$
- 性质：若 A, B 互不相容, 则 $P(A \cup B) = P(A) + P(B)$
- **对立事件**：若 A, B 互斥, 且 $P(A) + P(B) = 1$, 则称 A 和 B 为对立事件
- 举个例子：

设在某一时刻 zjm 去找 r****g 为事件 A，唱 hop 为事件 B，显然这两个事件互不相容 (这里默认他不会唱着 hop 去找 r****g)

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件
- 理解：A 和 B **不能同时发生**, 即 $P(AB) = 0$
- 性质：若 A, B 互不相容, 则 $P(A \cup B) = P(A) + P(B)$
- **对立事件**：若 A, B 互斥, 且 $P(A) + P(B) = 1$, 则称 A 和 B 为对立事件
- 举个例子：

设在某一时刻 zjm 去找 r****g 为事件 A，唱 hop 为事件 B，显然这两个事件互不相容 (这里默认他不会唱着 hop 去找 r****g)

那么, 他这一时刻找 r****g **或者**唱 hop 的概率为 $P(A \cup B)$= $P(A) + P(B)$

互不相容

- 定义：设 A, B 是两个事件, 且 $A \cap B = \varnothing$, 则称 A, B 为互斥事件, 也叫互不相容事件
- 理解：A 和 B **不能同时发生**, 即 $P(AB) = 0$
- 性质：若 A, B 互不相容, 则 $P(A \cup B) = P(A) + P(B)$
- **对立事件**：若 A, B 互斥, 且 $P(A) + P(B) = 1$, 则称 A 和 B 为对立事件
- 举个例子：

设在某一时刻 zjm 去找 r****g 为事件 A，唱 hop 为事件 B，显然这两个事件互不相容 (这里默认他不会唱着 hop 去找 r****g)

那么, 他这一时刻找 r****g **或者**唱 hop 的概率为 $P(A \cup B)= P(A) + P(B)$

如果 A 和 B 为对立事件，则说明他这一时刻~~一天到晚~~要么去找 r****g 要么唱 hop

相互独立

- 定义：设 A, B 是两个事件, 且 $P(AB) = P(A) \cdot P(B)$, 则称事件 A, B 相互独立

相互独立

- **定义：设 A, B 是两个事件, 且 $P(AB) = P(A) \cdot P(B)$, 则称事件 A, B 相互独立**
- 理解：
  1.A 发生和 B 发生**没有关系**, A B 的发生不会互相影响.

相互独立

- 定义：设 A, B 是两个事件, 且 $P(AB) = P(A) \cdot P(B)$, 则称事件 A, B 相互独立
- 理解：
  1.A 发生和 B 发生**没有关系**, A B 的发生不会互相影响.
  2. 若 $P(A) > 0$ 且 $P(B) > 0$, 则 A, B 相互独立与 A, B 互不相容不能同时成立

相互独立

- 定义：设 A,B 是两个事件, 且 $P(AB)=P(A)\cdot P(B)$, 则称事件 A,B 相互独立
- 理解：
  1.A 发生和 B 发生**没有关系**, A B 的发生不会互相影响.
  2. 若 $P(A)>0$ 且 $P(B)>0$, 则 A,B 相互独立与 A,B 互不相容不能同时成立
- 依旧举个例子：

相互独立

- 定义：设 A, B 是两个事件, 且 $P(AB) = P(A) \cdot P(B)$, 则称事件 A, B 相互独立
- 理解：
  1.A 发生和 B 发生**没有关系**, A B 的发生不会互相影响.
  2. 若 $P(A) > 0$ 且 $P(B) > 0$, 则 A, B 相互独立与 A, B 互不相容不能同时成立
- 依旧举个例子：

设 zjm 白天唱 hop 为事件 A，凌晨唱 hop 为事件 B，显然这两个事件相互独立

相互独立

- 定义：设 A, B 是两个事件, 且 $P(AB) = P(A) \cdot P(B)$, 则称事件 A, B 相互独立
- 理解：
  1.A 发生和 B 发生**没有关系**, A B 的发生不会互相影响.
  2. 若 $P(A) > 0$ 且 $P(B) > 0$, 则 A, B 相互独立与 A, B 互不相容不能同时成立
- 依旧举个例子：

设 zjm 白天唱 hop 为事件 A，凌晨唱 hop 为事件 B，显然这两个事件相互独立

那么白天唱 hop **且**凌晨唱 hop 的概率为 $P(AB) = P(A) \cdot P(B)$

相互独立

- 定义：设 A, B 是两个事件, 且 $P(AB) = P(A) \cdot P(B)$, 则称事件 A, B 相互独立
- 理解：
  1.A 发生和 B 发生**没有关系**, A B 的发生不会互相影响.
  2. 若 $P(A) > 0$ 且 $P(B) > 0$, 则 A, B 相互独立与 A, B 互不相容不能同时成立
- 依旧举个例子：

设 zjm 白天唱 hop 为事件 A，凌晨唱 hop 为事件 B，显然这两个事件相互独立

那么白天唱 hop **且**凌晨唱 hop 的概率为 $P(AB) = P(A) \cdot P(B)$

注意，此时他白天**或**凌晨唱 hop 的概率 $P(A \cup B) \neq P(A) + P(B)$，而应该等于 $P(A) + P(B) - P(AB)$

条件概率

定义：设 A, B 是两个事件, 且 $P(A) > 0$, 则称 $P(B|A) = \frac{P(AB)}{P(A)}$ 为在事件 A 发生的条件下事件 B 发生的条件概率

条件概率

定义：设 A, B 是两个事件, 且 $P(A) > 0$, 则称 $P(B|A) = \frac{P(AB)}{P(A)}$ 为在事件 A 发生的条件下事件 B 发生的条件概率

可列可加性：设 B, C 是两个互不相容的事件, 那么有

$$P(B \cup C\,|A) = P(B\,|A) + P(C\,|A)$$

条件概率

定义：设 A, B 是两个事件, 且 $P(A) > 0$, 则称 $P(B|A) = \frac{P(AB)}{P(A)}$ 为在事件 A 发生的条件下事件 B 发生的条件概率

可列可加性：设 B, C 是两个互不相容的事件, 那么有

$$P(B \cup C\ |A) = P(B\ |A) + P(C\ |A)$$

乘法公式：$P(AB) = P(A) \times P(B|A)$

全概率公式

划分：设样本空间 S, 若事件 $B_1, B_2, \ldots, B_n$ 两两互不相容且 $\bigcup_{i=1}^{n} B_i = S$, 则称 $B_1, B_2, \ldots, B_n$ 是样本空间 S 的一个划分

全概率公式

划分：设样本空间 S, 若事件 $B_1, B_2, \ldots, B_n$ 两两互不相容且 $\bigcup_{i=1}^{n} B_i = S$, 则称 $B_1, B_2, \ldots, B_n$ 是样本空间 S 的一个划分

定义：设 $B_1, B_2, \ldots, B_n$ 是样本空间的一个划分, 且 $\forall P(B_i) > 0$, 则

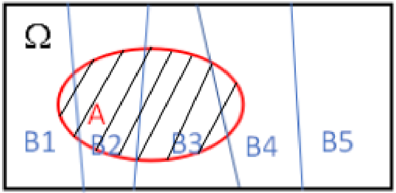
$$P(A) = \sum_{i=1}^{n} P(B_i) \times P(A|B_i)$$

全概率公式

划分：设样本空间 S, 若事件 $B_1, B_2, \ldots, B_n$ 两两互不相容且 $\bigcup_{i=1}^{n} B_i = S$, 则称 $B_1, B_2, \ldots, B_n$ 是样本空间 S 的一个划分

定义：设 $B_1, B_2, \ldots, B_n$ 是样本空间的一个划分, 且 $\forall P(B_i) > 0$, 则

$$P(A) = \sum_{i=1}^{n} P(B_i) \times P(A|B_i)$$



贝叶斯公式

定义：设 $B_1, B_2, \ldots, B_n$ **是样本空间的一个划分，且** $\forall P(B_i) > 0$**，则**

$$P(B_k|A) = \frac{P(B_kA)}{P(A)} = \frac{P(A|B_k) \times P(B_k)}{\sum_{i=1}^{n} P(B_i) \times P(A|B_i)}$$

贝叶斯公式

定义：设 $B_1, B_2, \ldots, B_n$ 是样本空间的一个划分, 且 $\forall P(B_i) > 0$, 则

$$P(B_k|A) = \frac{P(B_kA)}{P(A)} = \frac{P(A|B_k) \times P(B_k)}{\sum_{i=1}^{n} P(B_i) \times P(A|B_i)}$$

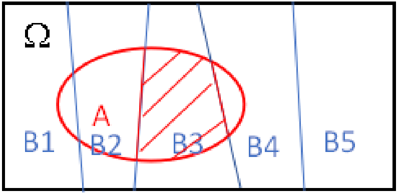
它实现了 $P(B_k|A)$ 与 $P(A|B_k)$ 的转化

贝叶斯公式

定义：设 $B_1, B_2, \ldots, B_n$ 是样本空间的一个划分，且 $\forall P(B_i) > 0$，则

$$P(B_k|A) = \frac{P(B_kA)}{P(A)} = \frac{P(A|B_k) \times P(B_k)}{\sum_{i=1}^{n} P(B_i) \times P(A|B_i)}$$

它实现了 $P(B_k|A)$ 与 $P(A|B_k)$ 的转化



小练习

机房里有 80% 的男生, 20% 的女生. 男生搞套路的概率为 5%, 女生搞套路的概率为 90%

(1) 若随机从机房中挑选一个人, 则搞套路的概率为多少？

(2) 从监控中随机抽取一个人, 发现 ta 在搞套路, 求 ta 是女生的概率为多少？

Solution

一个很简单的小练习，相信大家都会做

Solution

一个很简单的小练习，相信大家都会做

设 A 表示男生，B 表示搞套路

那么有 $P(A) = 80\%, P(\overline{A}) = 20\%$, $P(B|A) = 5\%, P(\overline{B}|A) = 95\%$, $P(B|\overline{A}) = 90\%, P(\overline{B}|\overline{A}) = 10\%$

Solution

一个很简单的小练习，相信大家都会做

设 A 表示男生，B 表示搞套路

那么有 $P(A)=80\%, P(\overline{A})=20\%,\ P(B|A)=5\%, P(\overline{B}|A)=95\%,$
$P(B|\overline{A})=90\%, P(\overline{B}|\overline{A})=10\%$

(1)$P(B)=P(A)\times P(B|A)+P(\overline{A})\times P(B|\overline{A})=22\%$

Solution

一个很简单的小练习，相信大家都会做

设 A 表示男生，B 表示搞套路

那么有 $P(A)=80\%, P(\overline{A})=20\%$, $P(B|A)=5\%, P(\overline{B}|A)=95\%$, $P(B|\overline{A})=90\%, P(\overline{B}|\overline{A})=10\%$

(1)$P(B)=P(A)\times P(B|A)+P(\overline{A})\times P(B|\overline{A})=22\%$

(2)$P(\overline{A}|B)=\frac{P(\overline{A})\times P(B|\overline{A})}{P(\overline{A})\times P(B|\overline{A})+P(A)\times P(B|A)}=82\%$

期望

对于一个离散型随机变量 x, 定义它的数学期望：

$$E(x) = \sum_{value} P(x = value) \times value$$

可以理解为把概率作为权值的加权平均数

期望

对于一个离散型随机变量 x, 定义它的数学期望：

$$E(x) = \sum_{value} P(x = value) \times value$$

可以理解为把概率作为权值的加权平均数

当随机变量 X, Y 独立时，$E(XY) = E(X) \times E(Y)$

期望的线性性

无论两个变量 X, Y 是否独立, 总有：

$$E(\alpha X+\beta Y)=\alpha E(X)+\beta E(Y)$$

期望的线性性

无论两个变量 X, Y 是否独立，总有：

$$E(\alpha X + \beta Y) = \alpha E(X) + \beta E(Y)$$

理解：和的期望等于期望的和

期望的线性性

无论两个变量 X, Y 是否独立，总有：

$$E(\alpha X + \beta Y) = \alpha E(X) + \beta E(Y)$$

理解：和的期望等于期望的和

例子：扔两次骰子点数的期望等于扔一次期望点数的两倍

期望的线性性

证明：

$$
\begin{aligned}
\alpha E(X)+\beta E(Y) =&\alpha\sum_i P(X=i)\times i+\beta\sum_j P(Y=j)\times j\\
=&\sum_i P(X=i)\times \alpha i+\sum_j P(Y=j)\times \beta j\\
=&\left(\sum_i P(X=i)\times \alpha i\right)\left(\sum_j P(Y=j)\right)+\\
&\left(\sum_j P(Y=j)\times \beta j\right)\left(\sum_i P(X=i)\right)\\
=&\sum_i\sum_j P(X=i,Y=j)\times \alpha i+\sum_i\sum_j P(X=i,Y=j)\times \beta j\\
=&\sum_i\sum_j P(X=i,Y=j)\times (\alpha i+\beta j)\\
=&E(\alpha X+\beta Y)
\end{aligned}
$$

全期望公式

$$E(X) = E(E(X|Y))$$

全期望公式

$$E(X) = E(E(X|Y))$$

理解：整体的平均值等于每个部分平均值的平均值

全期望公式

$$E(X) = E(E(X|Y))$$

理解：整体的平均值等于每个部分平均值的平均值

例子：全校的平均成绩既等于所有人总分除以总人数，又等于每个班的平均分除以班级数

全期望公式

证明：

$$
\begin{aligned}
E\left(E(X|Y)\right) &= \sum_j \left(\sum_i i \cdot P(X=i|Y=j)\right) \cdot P(Y=j) \\
&= \sum_j \sum_i i \cdot P(X=i|Y=j) \cdot P(Y=j) \\
&= \sum_i \sum_j \color{red}{i \cdot P(Y=j|X=i) \cdot P(X=i)} \\
&= \sum_i i \cdot P(X=i) \cdot \left(\sum_j P(Y=j|X=i)\right) \\
&= \sum_i i \cdot P(X=i) \\
&= E(X)
\end{aligned}
$$

方差

定义：

$$D(X)=E\big((X-E(X))^2\big)=E\big(X^2-2XE(X)+E^2(X)\big)=E(X^2)-E^2(X)$$

此时括号中的 $E(X)$ 看成一个常数

方差

定义：

$$D(X)=E\big((X-E(X))^2\big)=E\big(X^2-2XE(X)+E^2(X)\big)=E(X^2)-E^2(X)$$

此时括号中的 $E(X)$ **看成一个常数**

方差代表了随机变量的取值对于其数学期望的离散程度

常用技巧

- 概率顺推，期望倒推

常用技巧

- 概率顺推，期望倒推
- 利用期望线性性拆贡献

常用技巧

- 概率顺推，期望倒推
- 利用期望线性性拆贡献
- 方案数转化为概率期望

常用技巧

- 概率顺推，期望倒推
- 利用期望线性性拆贡献
- 方案数转化为概率期望
- 推式子，迭代

常用技巧

- 概率顺推，期望倒推
- 利用期望线性性拆贡献
- 方案数转化为概率期望
- 推式子，迭代
- 高斯消元，主元法

常用技巧

- 概率顺推, 期望倒推
- 利用期望线性性拆贡献
- 方案数转化为概率期望
- 推式子, 迭代
- 高斯消元, 主元法
- 赌徒破产模型

常用技巧

- 概率顺推，期望倒推
- 利用期望线性性拆贡献
- 方案数转化为概率期望
- 推式子，迭代
- 高斯消元，主元法
- 赌徒破产模型
- dp、容斥以及其他计数方面的技巧 (这不是此课件的重点)

期望的线性性

Description

给定一棵 n 个结点的树，你从点 S 出发开始 dfs 整棵树，走到一个叶子结点 T 时停止

每次 dfs 到一个点时随机遍历一个它相邻且未经过的结点

每新到一个点会使计数器加 1，回溯时也会加 1，求计数器值的期望

$n \leq 10^6$

Solution

根据期望的线性性，考虑每条边对答案的贡献

Solution

根据期望的线性性，考虑每条边对答案的贡献

将 S 到 T 的链提出来，并将这条链上的边称为关键边，在这条链上的点称为关键点

Solution

根据期望的线性性，考虑每条边对答案的贡献

将 S 到 T 的链提出来，并将这条链上的边称为关键边，在这条链上的点称为关键点

- 对于每条关键边，它一定并且只会经过一次，故它的贡献就为 1

Solution

根据期望的线性性，考虑每条边对答案的贡献

将 S 到 T 的链提出来，并将这条链上的边称为关键边，在这条链上的点称为关键点

- 对于每条关键边，它一定并且只会经过一次，故它的贡献就为 1
- 对于每条非关键边，它只可能经过 0 次或 2 次.

Solution

根据期望的线性性, 考虑每条边对答案的贡献

将 S 到 T 的链提出来, 并将这条链上的边称为关键边, 在这条链上的点称为关键点

- 对于每条关键边, 它一定并且只会经过一次, 故它的贡献就为 1
- 对于每条非关键边, 它只可能经过 0 次或 2 次.

Proof.

Solution

根据期望的线性性，考虑每条边对答案的贡献

将 S 到 T 的链提出来，并将这条链上的边称为关键边，在这条链上的点称为关键点

- 对于每条关键边，它一定并且只会经过一次，故它的贡献就为 1
- 对于每条非关键边，它只可能经过 0 次或 2 次.

Proof.

找到它最近的关键点祖先，那么它为 0 还是为 2 取决于这个关键点是先遍历关键边还是先遍历它所对应的那个关键点下的儿子

Solution

根据期望的线性性，考虑每条边对答案的贡献

将 S 到 T 的链提出来，并将这条链上的边称为关键边，在这条链上的点称为关键点

- 对于每条关键边，它一定并且只会经过一次，故它的贡献就为 1
- 对于每条非关键边，它只可能经过 0 次或 2 次.

Proof.

找到它最近的关键点祖先，那么它为 0 还是为 2 取决于这个关键点是先遍历关键边还是先遍历它所对应的那个关键点下的儿子

而这两种情况的概率均为 $\frac{1}{2}$，故这条边的贡献仍为 1. □

Solution

根据期望的线性性，考虑每条边对答案的贡献

将 S 到 T 的链提出来，并将这条链上的边称为关键边，在这条链上的点称为关键点

- 对于每条关键边，它一定并且只会经过一次，故它的贡献就为 1
- 对于每条非关键边，它只可能经过 0 次或 2 次.

Proof.

找到它最近的关键点祖先，那么它为 0 还是为 2 取决于这个关键点是先遍历关键边还是先遍历它所对应的那个关键点下的儿子

而这两种情况的概率均为 $\frac{1}{2}$，故这条边的贡献仍为 1. □

答案即为 $n-1$

Description

一开始有 n 个猎人，每个猎人在死后会随机选择一个场上存活的猎人开一枪将其杀死.

更具体地，每一个猎人有一个仇恨指数 w_i，每一轮猎人开枪的时候，每个存活的猎人被选中的概率均与其仇恨指数成正比.

现在你按照同样的方式随机开出了第一枪，接下来 n 轮之后所有猎人都会死，求 1 号猎人期望下会开第几枪时死掉.

$n \leq 10^5$

Solution

1 号猎人死亡的轮数等于在 1 号之前死亡的猎人数 $+1$

Solution

1 号猎人死亡的轮数等于在 1 号之前死亡的猎人数 $+1$

根据期望的线性性，转化成每个猎人比 1 号猎人先死的概率和，也就是需要求出 **i 号猎人比 1 号猎人先死的概率**

Solution

1 号猎人死亡的轮数等于在 1 号之前死亡的猎人数 $+1$

根据期望的线性性，转化成每个猎人比 1 号猎人先死的概率和，也就是需要求出 i **号猎人比** 1 **号猎人先死的概率**

原题每次概率的分母都在变化，不好计算，考虑把题目转化为：开枪时，每个猎人都能被打到. 如果打到死掉的猎人，就再打一枪，一直到打到活着的猎人停止

Solution

1 号猎人死亡的轮数等于在 1 号之前死亡的猎人数 $+1$

根据期望的线性性，转化成每个猎人比 1 号猎人先死的概率和，也就是需要求出 i **号猎人比** 1 **号猎人先死的概率**

原题每次概率的分母都在变化，不好计算，考虑把题目转化为：开枪时，每个猎人都能被打到．如果打到死掉的猎人，就再打一枪，一直到打到活着的猎人停止

Proof.

Solution

1 号猎人死亡的轮数等于在 1 号之前死亡的猎人数 $+1$

根据期望的线性性，转化成每个猎人比 1 号猎人先死的概率和，也就是需要求出 i **号猎人比** 1 **号猎人先死的概率**

原题每次概率的分母都在变化，不好计算，考虑把题目转化为：开枪时，每个猎人都能被打到．如果打到死掉的猎人，就再打一枪，一直到打到活着的猎人停止

Proof.

设 S 为所有人的 w_i 和，K 为当前已经死了的猎人的 w_i 和

Solution

1 号猎人死亡的轮数等于在 1 号之前死亡的猎人数 $+1$

根据期望的线性性，转化成每个猎人比 1 号猎人先死的概率和，也就是需要求出 i **号猎人比** 1 **号猎人先死的概率**

原题每次概率的分母都在变化，不好计算，考虑把题目转化为：开枪时，每个猎人都能被打到. 如果打到死掉的猎人，就再打一枪，一直到打到活着的猎人停止

Proof.

设 S 为所有人的 w_i 和，K 为当前已经死了的猎人的 w_i 和

转化后的题中，当前打到 i 号猎人的概率 $P = \frac{K}{S}P + \frac{w_i}{S}$

Solution

1 号猎人死亡的轮数等于在 1 号之前死亡的猎人数 $+1$

根据期望的线性性，转化成每个猎人比 1 号猎人先死的概率和，也就是需要求出 i **号猎人比** 1 **号猎人先死的概率**

原题每次概率的分母都在变化，不好计算，考虑把题目转化为：开枪时，每个猎人都能被打到．如果打到死掉的猎人，就再打一枪，一直到打到活着的猎人停止

Proof.

设 S 为所有人的 w_i 和，K 为当前已经死了的猎人的 w_i 和

转化后的题中，当前打到 i 号猎人的概率 $P = \frac{K}{S}P + \frac{w_i}{S}$

化简得到 $P = \frac{w_i}{S-K}$，与原题相等 $\square$

Solution

于是 i 号猎人比 1 号猎人先死的概率为

$$\sum_{j=0}^{\infty}\left(\frac{S-w_1-w_i}{S}\right)^j \cdot \frac{w_i}{S}$$

Solution

于是 i 号猎人比 1 号猎人先死的概率为

$$\sum_{j=0}^{\infty}(\frac{S-w_1-w_i}{S})^j \cdot \frac{w_i}{S}$$

等比数列求和后等于 $\dfrac{w_i}{w_i+w_1}$

Solution

于是 i 号猎人比 1 号猎人先死的概率为

$$\sum_{j=0}^{\infty}\left(\frac{S-w_1-w_i}{S}\right)^j\cdot\frac{w_i}{S}$$

等比数列求和后等于 $\dfrac{w_i}{w_i+w_1}$

答案即为 $\displaystyle\sum_{i=2}^{n}\frac{w_i}{w_i+w_1}$

Description

有 n 个物品，第 i 个物品价值为 a_i. 对这 n 个物品执行以下操作：

- 选择一个未被删除的物品，将其删除. 操作的代价为这个物品所在联通块内的权值和 (联通的定义是位置相邻且物品未被删除)

求所有删除物品方案的代价和

$n \le 10^5$

Solution

直接组合计算答案很不好算，考虑转化为总方案数乘**每个物品对答案贡献的期望**，也就是要求第 i 个物品**产生贡献的概率**

Solution

直接组合计算答案很不好算，考虑转化为总方案数乘**每个物品对答案贡献的期望**，也就是要求第 i 个物品**产生贡献的概率**

根据期望的线性性，对于第 i 个物品，我们从 1 到 n 枚举物品 j，计算在删 j 时，i 与 j 联通的概率

Solution

直接组合计算答案很不好算，考虑转化为总方案数乘**每个物品对答案贡献的期望**，也就是要求第 i 个物品**产生贡献的概率**

根据期望的线性性，对于第 i 个物品，我们从 1 到 n 枚举物品 j，计算在删 j 时，i 与 j 联通的概率

不难发现，这个概率为 $\frac{1}{|i-j|+1}$

Solution

直接组合计算答案很不好算, 考虑转化为总方案数乘**每个物品对答案贡献的期望**, 也就是要求第 i 个物品**产生贡献的概率**

根据期望的线性性, 对于第 i 个物品, 我们从 1 到 n 枚举物品 j, 计算在删 j 时, i 与 j 联通的概率

不难发现, 这个概率为 $\frac{1}{|i-j|+1}$

Proof.

Solution

直接组合计算答案很不好算，考虑转化为总方案数乘**每个物品对答案贡献的期望**，也就是要求第 i 个物品**产生贡献的概率**

根据期望的线性性，对于第 i 个物品，我们从 1 到 n 枚举物品 j，计算在删 j 时，i 与 j 联通的概率

不难发现，这个概率为 $\frac{1}{|i-j|+1}$

Proof.

假设 $i < j$，则 j 需要满足，它是 $[i, j]$ 这些物品中最先被删除的

Solution

直接组合计算答案很不好算，考虑转化为总方案数乘**每个物品对答案贡献的期望**，也就是要求第 i 个物品**产生贡献的概率**

根据期望的线性性，对于第 i 个物品，我们从 1 到 n 枚举物品 j，计算在删 j 时，i 与 j 联通的概率

不难发现，这个概率为 $\frac{1}{|i-j|+1}$

Proof.

假设 $i < j$，则 j 需要满足，它是 $[i, j]$ 这些物品中最先被删除的

也就是在只考虑 $[i, j]$ 这些物品的任意排列方案时，**钦定第一个为 j 的概率**，即为 $\frac{1}{|i-j|+1}$ □

Solution

直接组合计算答案很不好算，考虑转化为总方案数乘**每个物品对答案贡献的期望**，也就是要求第 i 个物品**产生贡献的概率**

根据期望的线性性，对于第 i 个物品，我们从 1 到 n 枚举物品 j，计算在删 j 时，i 与 j 联通的概率

不难发现，这个概率为 $\frac{1}{|i-j|+1}$

Proof.

假设 $i<j$，则 j 需要满足，它是 $[i,j]$ 这些物品中最先被删除的

也就是在只考虑 $[i,j]$ 这些物品的任意排列方案时，**钦定第一个为 j 的概率**，即为 $\frac{1}{|i-j|+1}$ □

答案即为 $\sum_{i=1}^{n} a_i \sum_{j=1}^{n} \frac{1}{|i-j|+1}$

Description

给你一张 n 个点 m 条边的图

每次随机选取一个点，将答案加上这个点所在的联通块大小并删去该点

求答案的期望，对 998244353 取模

$n \le 10^5, m \in [n-1, n]$

Solution

先考虑树的情况：

Solution

先考虑树的情况：

根据期望的线性性，我们可以考虑**两两点对**对答案的贡献

Solution

先考虑树的情况：

根据期望的线性性，我们可以考虑**两两点对**对答案的贡献

通过上一题我们知道，对于有序点对 (x, y)，树上路径长度为 c(经过 c 个节点)，对答案的贡献为 $\frac{1}{c}$

Solution

先考虑树的情况：

根据期望的线性性，我们可以考虑**两两点对**对答案的贡献

通过上一题我们知道，对于有序点对 (x, y)，树上路径长度为 c(经过 c 个节点)，对答案的贡献为 $\frac{1}{c}$

于是我们只需要对每个 c 求出长度为 c 的路径条数

点分治 $+$ NTT 合并即可

Solution

先考虑树的情况：

根据期望的线性性，我们可以考虑**两两点对**对答案的贡献

通过上一题我们知道，对于有序点对 (x, y)，树上路径长度为 c(经过 c 个节点)，对答案的贡献为 $\frac{1}{c}$

于是我们只需要对每个 c 求出长度为 c 的路径条数

点分治 $+$ NTT 合并即可

接下来考虑基环树，仍然是考虑点对 (x, y) 的贡献：

Solution

先考虑树的情况：

根据期望的线性性，我们可以考虑**两两点对**对答案的贡献

通过上一题我们知道，对于有序点对 (x, y)，树上路径长度为 c(经过 c 个节点)，对答案的贡献为 $\frac{1}{c}$

于是我们只需要对每个 c 求出长度为 c 的路径条数

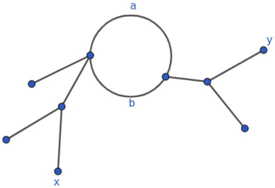
点分治 $+$ NTT 合并即可

接下来考虑基环树，仍然是考虑点对 (x, y) 的贡献：

- (x, y) 在同一棵子树内：与树的计算方法相同

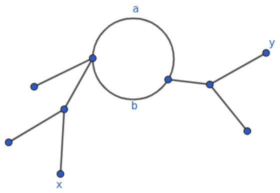
Solution

- (x, y) 不在同一棵子树内：



Solution

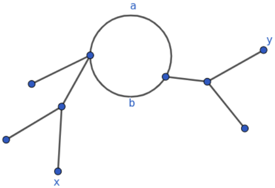
- (x, y) 不在同一棵子树内：



令 $dep[x] + dep[y] = c$, 则此时删除 x 时 x, y 仍联通的概率等于 x 在 $a + c$ 个点中最先删除**或**在 $b + c$ 个点中最先删除的概率 (分别对应经过环的两侧)

Solution

- (x, y) 不在同一棵子树内：

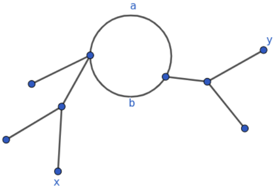


令 $dep[x] + dep[y] = c$, 则此时删除 x 时 x, y 仍联通的概率等于 x 在 $a + c$ 个点中最先删除**或**在 $b + c$ 个点中最先删除的概率 (分别对应经过环的两侧)

通过概率的加法公式 $P(A \cup B) = P(A) + P(B) - P(AB)$ 我们可以得到此时贡献为 $\frac{1}{a+c} + \frac{1}{b+c} - \frac{1}{a+b+c}$

Solution

- (x, y) 不在同一棵子树内：



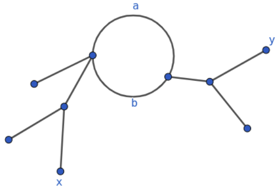
令 $dep[x] + dep[y] = c$，则此时删除 x 时 x, y 仍联通的概率等于 x 在 $a + c$ 个点中最先删除**或**在 $b + c$ 个点中最先删除的概率 (分别对应经过环的两侧)

通过概率的加法公式 $P(A \cup B) = P(A) + P(B) - P(AB)$ 我们可以得到此时贡献为 $\frac{1}{a+c} + \frac{1}{b+c} - \frac{1}{a+b+c}$

随便选一条边破环为链，用类似 CDQ 分治的分治方法 + NTT 即可

Solution

- (x, y) 不在同一棵子树内：



令 $dep[x] + dep[y] = c$，则此时删除 x 时 x, y 仍联通的概率等于 x 在 $a + c$ 个点中最先删除**或**在 $b + c$ 个点中最先删除的概率 (分别对应经过环的两侧)

通过概率的加法公式 $P(A \cup B) = P(A) + P(B) - P(AB)$ 我们可以得到此时贡献为 $\frac{1}{a+c} + \frac{1}{b+c} - \frac{1}{a+b+c}$

随便选一条边破环为链，用类似 CDQ 分治的分治方法 + NTT 即可

具体实现可见此

简单推式子

Description

xz 被 n 个彪形大汉围了起来, 他可以去找其中的某个大汉, 请求放过他. 而每个大汉有一个属性值 t

- 若 $t < 0$, 那么这个大汉其实是 zqc 小解杰变的, 她会花费 t 的时间带他逃离人群
- 若 $t > 0$, 那么这个大汉其实是 zjm 大葛♂阁变的, 他会花费 t 的时间揍 xz♂一顿, 让他回到原地, 并且失去之前的所有记忆

已知 xz 选择每个大汉的概率是相同的, 求他逃离人群花费时间的期望. 若无法逃离, 则输出 inf

$n \leq 10^5$

Solution

很简单的一道题，首先考虑一次事件的结果：

Solution

很简单的一道题，首先考虑一次事件的结果：

- 逃离人群：期望花费 $\frac{t}{n}$

Solution

很简单的一道题，首先考虑一次事件的结果：

- 逃离人群：期望花费 $\frac{t}{n}$
- 揍一顿：期望花费 $\frac{t+E}{n}$
  其中 E 表示逃离人群的期望花费

Solution

很简单的一道题，首先考虑一次事件的结果：

- 逃离人群：期望花费 $\frac{t}{n}$
- 揍一顿：期望花费 $\frac{t+E}{n}$
  其中 E 表示逃离人群的期望花费

设 zqc 小姐姐有 k_1 个，zjm 大哥哥有 k_2，逃离人群的时间总和为 s_1，揍一顿的时间总和为 s_2，那么

$$
\begin{aligned}
E &= \frac{s_1}{n} + \frac{s_2 + k_2 E}{n} \\
&= \frac{s_1 + s_2}{n - k_2}
\end{aligned}
$$

Solution

很简单的一道题，首先考虑一次事件的结果：

- 逃离人群：期望花费 $\frac{t}{n}$
- 揍一顿：期望花费 $\frac{t+E}{n}$
  其中 E 表示逃离人群的期望花费

设 zqc 小姐姐有 k_1 个，zjm 大哥哥有 k_2，逃离人群的时间总和为 s_1，揍一顿的时间总和为 s_2，那么

$$
\begin{aligned}
E &= \frac{s_1}{n} + \frac{s_2 + k_2 E}{n} \\
&= \frac{s_1 + s_2}{n - k_2}
\end{aligned}
$$

当 $k_2 = n$ 时无解

Description

给出正整数 p_a, p_b, k, 满足 $p_a + p_b = 1$

一开始有一个空串, 每一次有 p_a 的概率在末尾加入 a, p_b 的概率在串末尾加入 b

当串中存在 k 个子序列 ab 时停止, 求停止时子串长度的期望, 答案对 $10^9 + 7$ 取模

$k \leq 1000$

Solution

首先有一个很简单的 dp：

Solution

首先有一个很简单的 dp：设 $dp[i][j]$ 表示当前已经有 i 个 a, j 个子序列 ab, 到结束时 ab 个数的期望, 那么有

$$dp[i][j] = p_a * dp[i+1][j] + p_b * dp[i][i+j]$$

Solution

首先有一个很简单的 dp：设 $dp[i][j]$ 表示当前已经有 i 个 a, j 个子序列 ab, 到结束时 ab 个数的期望, 那么有

$$dp[i][j] = p_a * dp[i+1][j] + p_b * dp[i][i+j]$$

但是这个 dp 的边界情况很不好考虑

Solution

首先有一个很简单的 dp：设 $dp[i][j]$ 表示当前已经有 i 个 a, j 个子序列 ab, 到结束时 ab 个数的期望, 那么有

$$dp[i][j] = p_a * dp[i+1][j] + p_b * dp[i][i+j]$$

但是这个 dp 的边界情况很不好考虑

当 $i + j \geq k$ 时, 只要再加一个 b 就能停止, 于是我们可以枚举这个 b 什么时候加

Solution

首先有一个很简单的 dp：设 $dp[i][j]$ 表示当前已经有 i 个 a, j 个子序列 ab, 到结束时 ab 个数的期望, 那么有

$$dp[i][j] = p_a * dp[i+1][j] + p_b * dp[i][i+j]$$

但是这个 dp 的边界情况很不好考虑

当 $i + j \geq k$ 时, 只要再加一个 b 就能停止, 于是我们可以枚举这个 b 什么时候加

即当 $i + j \geq k$ 时, $dp[i][j] = p_b \sum_{x=0}^{\infty} (i + j + x) p_a^x$

Solution

首先有一个很简单的 dp：设 $dp[i][j]$ 表示当前已经有 i 个 a, j 个子序列 ab, 到结束时 ab 个数的期望, 那么有

$$dp[i][j] = p_a * dp[i+1][j] + p_b * dp[i][i+j]$$

但是这个 dp 的边界情况很不好考虑

当 $i + j \geq k$ 时, 只要再加一个 b 就能停止, 于是我们可以枚举这个 b 什么时候加

即当 $i + j \geq k$ 时, $dp[i][j] = p_b \sum_{x=0}^{\infty} (i + j + x) p_a^x$

等比数列求和一下得到 $dp[i][j] = i + j + \frac{p_a}{p_b}$

允许误差

Description

有一棵树，初始只有一个节点，q 次操作:

- 1 fa，新增一个节点，其父亲为 fa
- 2 x，询问：如果将 x 的子树内的每条边有 $\frac{1}{2}$ 的概率被删掉，该子树的深度的期望是多少 (注意询问不会真的删掉边)

$q \leq 5 * 10^5$，保留六位小数

Solution

一个直观想法是设 $dp[x][i]$ 表示 x 子树中, 随机删边后深度为 i 的概率

Solution

一个直观想法是设 $dp[x][i]$ 表示 x 子树中, 随机删边后深度为 i 的概率

但是因为子树深度是对所有儿子子树深度取 $\max$, 如果状态设" 深度为 i 的概率" 的话不好保证在哪里取到这个最大值

Solution

一个直观想法是设 $dp[x][i]$ 表示 x 子树中，随机删边后深度为 i 的概率

但是因为子树深度是对所有儿子子树深度取 $\max$，如果状态设" 深度为 i 的概率" 的话不好保证在哪里取到这个最大值

于是状态改为 $dp[x][i]$ 表示 x 子树中，随机删边后深度 $\leq i$ 的概率

Solution

一个直观想法是设 $dp[x][i]$ 表示 x 子树中，随机删边后深度为 i 的概率

但是因为子树深度是对所有儿子子树深度取 $\max$，如果状态设" 深度为 i 的概率" 的话不好保证在哪里取到这个最大值

于是状态改为 $dp[x][i]$ 表示 x 子树中，随机删边后深度 $\leq i$ 的概率

因为子树互相独立，所以概率可以直接乘

$$dp[x][i] = \prod_{y \in child(x)} 0.5 \times (dp[y][i-1] + 1)$$

Solution

一个直观想法是设 $dp[x][i]$ 表示 x 子树中，随机删边后深度为 i 的概率

但是因为子树深度是对所有儿子子树深度取 $\max$，如果状态设" 深度为 i 的概率" 的话不好保证在哪里取到这个最大值

于是状态改为 $dp[x][i]$ 表示 x 子树中，随机删边后深度 $\leq i$ 的概率

因为子树互相独立，所以概率可以直接乘

$$dp[x][i] = \prod_{y \in child(x)} 0.5 \times (dp[y][i-1] + 1)$$

深度较大时期望值很小，因为允许精度误差所以可以忽略. 加入每个点时把上面 50 个祖先的 dp 值更新一下即可

Description

有一棵 n 个点的树, 由如下方式生成:

- 根为 1, 第 $i(i > 1)$ 个点的父亲为 $[1, i-1]$ 的均匀随机整数.

定义一棵树的权值为 $\sum_{i=1}^{n} deg_i^2$(deg 是度数), 求这棵树权值的期望

$1 \leq n \leq 10^6$, 保留六位小数

Solution

对于每种度数的点统计个数，设 $dp[i][j]$ 表示 i 个点的树，度数为 j 的点的期望个数

Solution

对于每种度数的点统计个数，设 $dp[i][j]$ 表示 i 个点的树，度数为 j 的点的期望个数

只有父亲的度数为 $j-1$ 或 j 时才会对 $dp[i][j]$ 产生影响

Solution

对于每种度数的点统计个数，设 $dp[i][j]$ 表示 i 个点的树，度数为 j 的点的期望个数

只有父亲的度数为 $j-1$ 或 j 时才会对 $dp[i][j]$ 产生影响

$$dp[i][j] = dp[i-1][j] + \frac{dp[i-1][j-1]}{i-1} - \frac{dp[i-1][j]}{i-1} + [j=1]$$

Solution

对于每种度数的点统计个数，设 $dp[i][j]$ 表示 i 个点的树，度数为 j 的点的期望个数

只有父亲的度数为 $j-1$ 或 j 时才会对 $dp[i][j]$ 产生影响

$$dp[i][j] = dp[i-1][j] + \frac{dp[i-1][j-1]}{i-1} - \frac{dp[i-1][j]}{i-1} + [j=1]$$

度数较大时期望值很小，因此只需要跑 $j \le 50$ 的状态即可

普通高斯消元

Description

给出一张 n 个点 m 条边的无向图，其中某些点是关键点，1 号点一定不是关键点，n 号点一定是关键点

求从 1 开始走，每次随机选择一个相邻的点走过去，经过恰好 k 个关键点到达 n 的概率.

$n \le 500, m \le 500000, k \le 10^9, a \le 100$, 其中 a 表示关键点数量

Solution

设 $f[u][v]$ 为, 从 u 开始游走, 碰到的第一个关键点是 v 的概率

Solution

设 $f[u][v]$ 为, 从 u 开始游走, 碰到的第一个关键点是 v 的概率

假设求出了 $f[u][v]$, 那么设 $dp[i][j]$ 为遇到的第 i 个关键点为 j 的概率, $dp[k-1][n]$ 即为答案. 显然这一部分可以用矩阵快速幂优化

Solution

设 $f[u][v]$ 为, 从 u 开始游走, 碰到的第一个关键点是 v 的概率

假设求出了 $f[u][v]$, 那么设 $dp[i][j]$ 为遇到的第 i 个关键点为 j 的概率, $dp[k-1][n]$ 即为答案. 显然这一部分可以用矩阵快速幂优化

接下来考虑考虑求 $f[u][v]$, 一个暴力的想法是对于每个 v, 类似 [HNOI2013] 游走, 暴力高斯消元计算 $f[u][v]$

Solution

设 $f[u][v]$ 为, 从 u 开始游走, 碰到的第一个关键点是 v 的概率

假设求出了 $f[u][v]$, 那么设 $dp[i][j]$ 为遇到的第 i 个关键点为 j 的概率, $dp[k-1][n]$ 即为答案. 显然这一部分可以用矩阵快速幂优化

接下来考虑考虑求 $f[u][v]$, 一个暴力的想法是对于每个 v, 类似 [HNOI2013] 游走, 暴力高斯消元计算 $f[u][v]$

但是这样复杂度就上升到 $O(n^4)$, 无法接受

Solution

设 $f[u][v]$ 为, 从 u 开始游走, 碰到的第一个关键点是 v 的概率

假设求出了 $f[u][v]$, 那么设 $dp[i][j]$ 为遇到的第 i 个关键点为 j 的概率, $dp[k-1][n]$ 即为答案. 显然这一部分可以用矩阵快速幂优化

接下来考虑考虑求 $f[u][v]$, 一个暴力的想法是对于每个 v, 类似 [HNOI2013] 游走, 暴力高斯消元计算 $f[u][v]$

但是这样复杂度就上升到 $O(n^4)$, 无法接受

通过观察发现, 对于不同的 v 在高斯消元时, 左边的系数矩阵都没有变化, 只有右边的矩阵发生了变化. 于是可以 $O(n^3)$ 预处理出系数矩阵的逆矩阵, 每次 $O(n^2)$ 矩阵乘法即可

Solution

设 $f[u][v]$ 为, 从 u 开始游走, 碰到的第一个关键点是 v 的概率

假设求出了 $f[u][v]$, 那么设 $dp[i][j]$ 为遇到的第 i 个关键点为 j 的概率, $dp[k-1][n]$ 即为答案. 显然这一部分可以用矩阵快速幂优化

接下来考虑考虑求 $f[u][v]$, 一个暴力的想法是对于每个 v, 类似 [HNOI2013] 游走, 暴力高斯消元计算 $f[u][v]$

但是这样复杂度就上升到 $O(n^4)$, 无法接受

通过观察发现, 对于不同的 v 在高斯消元时, 左边的系数矩阵都没有变化, 只有右边的矩阵发生了变化. 于是可以 $O(n^3)$ 预处理出系数矩阵的逆矩阵, 每次 $O(n^2)$ 矩阵乘法即可

时间复杂度 $O(n^3 + a^3 \log k)$

主元法

Brief Introduction

这类题在列出 dp 式后，往往比较难直接递推，暴力高斯消元复杂度又比较高

Brief Introduction

这类题在列出 dp 式后，往往比较难直接递推，暴力高斯消元复杂度又比较高

可以把某个边界 dp 值看作主元 x，然后把其他 dp 值用它来表示 (表示成 $ax+b$ 的形式，维护系数 a,b)，再利用一些等量关系 (边界条件) 解出 x

Description

有一个 m 面的骰子，不停地掷骰子，求以下两种情况的期望步数：

1. 连续出现 n 个相同的时候停止
2. 连续出现 n 个不同的时候停止

$n, m \leq 10^6$

Solution

首先很容易设出一个 dp：$f[i]$ 表示连续出现 i 个相同，到停止时的期望步数，$g[i]$ 表示连续出现 i 个不同，到停止时的期望步数

$$f[i] = \frac{1}{m}f[i+1] + \frac{m-1}{m}f[1] + 1$$

$$g[i] = \frac{m-i}{m}g[i+1] + \frac{1}{m}\sum_{j=1}^{i} g[j] + 1$$

Solution

首先很容易设出一个 dp：$f[i]$ 表示连续出现 i 个相同，到停止时的期望步数，$g[i]$ 表示连续出现 i 个不同，到停止时的期望步数

$$f[i] = \frac{1}{m}f[i+1] + \frac{m-1}{m}f[1] + 1$$

$$g[i] = \frac{m-i}{m}g[i+1] + \frac{1}{m}\sum_{j=1}^{i} g[j] + 1$$

将 $f[1], g[1]$ 看成主元直接做就行了

Solution

首先很容易设出一个 dp：$f[i]$ 表示连续出现 i 个相同，到停止时的期望步数，$g[i]$ 表示连续出现 i 个不同，到停止时的期望步数

$$f[i] = \frac{1}{m} f[i+1] + \frac{m-1}{m} f[1] + 1$$

$$g[i] = \frac{m-i}{m} g[i+1] + \frac{1}{m} \sum_{j=1}^{i} g[j] + 1$$

将 $f[1], g[1]$ 看成主元直接做就行了

下面以 f 为例，稍微讲一下具体过程

Solution

首先很容易设出一个 dp：$f[i]$ 表示连续出现 i 个相同，到停止时的期望步数，$g[i]$ 表示连续出现 i 个不同，到停止时的期望步数

$$f[i] = \frac{1}{m}f[i+1] + \frac{m-1}{m}f[1] + 1$$

$$g[i] = \frac{m-i}{m}g[i+1] + \frac{1}{m}\sum_{j=1}^{i} g[j] + 1$$

将 $f[1], g[1]$ 看成主元直接做就行了

下面以 f 为例，稍微讲一下具体过程

原式化简得到 $f[i+1] = mf[i] - (m-1)f[1] - 1$

Solution

首先很容易设出一个 dp：$f[i]$ 表示连续出现 i 个相同，到停止时的期望步数，$g[i]$ 表示连续出现 i 个不同，到停止时的期望步数

$$f[i] = \frac{1}{m}f[i+1] + \frac{m-1}{m}f[1] + 1$$

$$g[i] = \frac{m-i}{m}g[i+1] + \frac{1}{m}\sum_{j=1}^{i} g[j] + 1$$

将 $f[1], g[1]$ 看成主元直接做就行了

下面以 f 为例，稍微讲一下具体过程

原式化简得到 $f[i+1] = mf[i] - (m-1)f[1] - 1$

当 $i = 1$ 时，$f[2] = f[1] - m$；当 $i > 1$ 时，直接代进去算即可

Solution

首先很容易设出一个 dp：$f[i]$ 表示连续出现 i 个相同，到停止时的期望步数，$g[i]$ 表示连续出现 i 个不同，到停止时的期望步数

$$f[i] = \frac{1}{m}f[i+1] + \frac{m-1}{m}f[1] + 1$$

$$g[i] = \frac{m-i}{m}g[i+1] + \frac{1}{m}\sum_{j=1}^{i} g[j] + 1$$

将 $f[1], g[1]$ 看成主元直接做就行了

下面以 f 为例，稍微讲一下具体过程

原式化简得到 $f[i+1] = mf[i] - (m-1)f[1] - 1$

当 $i = 1$ 时，$f[2] = f[1] - m$；当 $i > 1$ 时，直接代进去算即可

利用 $f[n] = 0$ 这个方程即能求出 $f[1]$ 的值

Description

$n \times m$ 的棋盘, 上面有一个骑士, 每个回合, 他会随机走马步 (一维坐标变化 1, 另一位坐标变化 2), 选择 8 种走马步的方式的概率都是给定的. 走出棋盘则游戏结束.

求对于每个初始位置, 期望进行多少个回合才会走出棋盘.

$n, m \leq 200$

Solution

设 $dp[x][y]$ 表示当前在 (x, y) 位置期望会在多少个回合后走出棋盘，转移很显然，但直接高消复杂度太大

Solution

设 $dp[x][y]$ 表示当前在 (x,y) 位置期望会在多少个回合后走出棋盘，转移很显然，但直接高消复杂度太大

考虑把前两行和第一列的状态设为主元，那么可以利用 $dp[i][j]$ 的方程把 $dp[i+2][j+1]$ 用主元表示

Solution

设 $dp[x][y]$ 表示当前在 (x, y) 位置期望会在多少个回合后走出棋盘，转移很显然，但直接高消复杂度太大

考虑把前两行和第一列的状态设为主元，那么可以利用 $dp[i][j]$ 的方程把 $dp[i+2][j+1]$ 用主元表示

如果 $(i+2, j+1)$ 在棋盘外，则 $dp[i+2][j+1] = 0$，这样的方程有主元个数个

Solution

设 $dp[x][y]$ 表示当前在 (x, y) 位置期望会在多少个回合后走出棋盘，转移很显然，但直接高消复杂度太大

考虑把前两行和第一列的状态设为主元，那么可以利用 $dp[i][j]$ 的方程把 $dp[i+2][j+1]$ 用主元表示

如果 $(i+2, j+1)$ 在棋盘外，则 $dp[i+2][j+1] = 0$，这样的方程有主元个数个

根据这些方程高消解方程即可，复杂度 $O((n+m)^3)$

Description

给定一棵 n 个点的树, 点权为 $0/1$, 首先随机一个点作为起点, 然后每次随机一个点 (可能和之前所在点相同) 走到该点并将其点权异或上 1

求期望的移动距离使得所有点点权相同.

$n \leq 10^5$

Solution

根据期望的线性性，考虑每个点对答案的贡献.

Solution

根据期望的线性性，考虑每个点对答案的贡献.

每次选择了一个点 i 后，**如果没有到达目标状态**，那么下一步期望的移动距离就是 i 到所有点路径长度的平均值 $\overline{w_i}$

Solution

根据期望的线性性, 考虑每个点对答案的贡献.

每次选择了一个点 i 后, **如果没有到达目标状态**, 那么下一步期望的移动距离就是 i 到所有点路径长度的平均值 $\overline{w_i}$

对于每个点, 我们只需要求出它在**没有到达目标状态前**期望会被经过多少次, 再乘上 $\overline{w_i}$, 就是 i 的贡献了

Solution

根据期望的线性性，考虑每个点对答案的贡献.

每次选择了一个点 i 后，**如果没有到达目标状态**，那么下一步期望的移动距离就是 i 到所有点路径长度的平均值 $\overline{w_i}$

对于每个点，我们只需要求出它在**没有到达目标状态前**期望会被经过多少次，再乘上 $\overline{w_i}$，就是 i 的贡献了

显然，树的形态并不影响点被经过的次数，在 0 和 1 的个数确定的情况下，**点权相同的点**期望被经过的次数是相同的

Solution

设 $dp[i][0/1]$ 表示有 i 个 1 的局面下, **某一个**权值为 $0/1$ 的点的期望经过次数, 那么有

$$dp[i][0] = \frac{i}{n}dp[i-1][0] + \frac{n-i-1}{n}dp[i+1][0] + \frac{1}{n}(dp[i+1][1]+1)$$
$$dp[i][1] = \frac{n-i}{n}dp[i+1][1] + \frac{i-1}{n}dp[i-1][1] + \frac{1}{n}(dp[i-1][0]+1)$$

其中 $dp[0][0/1] = dp[n][0/1] = 0$

Solution

设 $dp[i][0/1]$ 表示有 i 个 1 的局面下, **某一个**权值为 $0/1$ 的点的期望经过次数, 那么有

$$dp[i][0] = \frac{i}{n}dp[i-1][0] + \frac{n-i-1}{n}dp[i+1][0] + \frac{1}{n}(dp[i+1][1]+1)$$
$$dp[i][1] = \frac{n-i}{n}dp[i+1][1] + \frac{i-1}{n}dp[i-1][1] + \frac{1}{n}(dp[i-1][0]+1)$$

其中 $dp[0][0/1] = dp[n][0/1] = 0$

移项后可以发现, 根据 $dp[i][0/1]$ 可以推出 $dp[i+1][1]$, 根据 $dp[i][0/1]$ 和 $dp[i+1][1]$ 可以推出 $dp[i+1][0]$

Solution

设 $dp[i][0/1]$ 表示有 i 个 1 的局面下, **某一个**权值为 $0/1$ 的点的期望经过次数, 那么有

$$dp[i][0] = \frac{i}{n}dp[i-1][0] + \frac{n-i-1}{n}dp[i+1][0] + \frac{1}{n}(dp[i+1][1]+1)$$
$$dp[i][1] = \frac{n-i}{n}dp[i+1][1] + \frac{i-1}{n}dp[i-1][1] + \frac{1}{n}(dp[i-1][0]+1)$$

其中 $dp[0][0/1] = dp[n][0/1] = 0$

移项后可以发现, 根据 $dp[i][0/1]$ 可以推出 $dp[i+1][1]$, 根据 $dp[i][0/1]$ 和 $dp[i+1][1]$ 可以推出 $dp[i+1][0]$

考虑把 $dp[1][0/1]$ 设为主元, 最后根据在 $i = n-1$ 时的式子列出一个二元一次方程组, 解之即可

Solution

$$dp[i][0] = \frac{i}{n}dp[i-1][0] + \frac{n-i-1}{n}dp[i+1][0] + \frac{1}{n}(dp[i+1][1]+1)$$
$$dp[i][1] = \frac{n-i}{n}dp[i+1][1] + \frac{i-1}{n}dp[i-1][1] + \frac{1}{n}(dp[i-1][0]+1)$$

需要注意的地方：

Solution

$$dp[i][0] = \frac{i}{n}dp[i-1][0] + \frac{n-i-1}{n}dp[i+1][0] + \frac{1}{n}(dp[i+1][1]+1)$$
$$dp[i][1] = \frac{n-i}{n}dp[i+1][1] + \frac{i-1}{n}dp[i-1][1] + \frac{1}{n}(dp[i-1][0]+1)$$

需要注意的地方：

- 对于 $dp[n-1][0], dp[1][1]$, 式子 $\frac{1}{n}(dp[][]+1)$ 那一整块都不要加. 这是因为, 如果走回自己则到达目标状态, 不会再往下走, 因此期望步数无需 $+\frac{1}{n}$

Solution

$$dp[i][0] = \frac{i}{n}dp[i-1][0] + \frac{n-i-1}{n}dp[i+1][0] + \frac{1}{n}(dp[i+1][1]+1)$$
$$dp[i][1] = \frac{n-i}{n}dp[i+1][1] + \frac{i-1}{n}dp[i-1][1] + \frac{1}{n}(dp[i-1][0]+1)$$

需要注意的地方：

- 对于 $dp[n-1][0], dp[1][1]$, 式子 $\frac{1}{n}(dp[][]+1)$ 那一整块都不要加. 这是因为, 如果走回自己则到达目标状态, 不会再往下走, 因此期望步数无需 $+\frac{1}{n}$
- 最终的期望经过次数应还需加上 $\frac{1}{n}$, 因为一开始随机选一个点的那一步没有统计在 dp 中

min-max 容斥

Brief Introduction

~~相信大家都会这个东西~~

不会的同学可以去问 zqc ~~小解杰~~

Brief Introduction

~~相信大家都会这个东西~~

不会的同学可以去问 zqc ~~小解杰~~

$$\max(S) = \sum_{T \in S, T \neq \varnothing} (-1)^{|T|+1} \min(T)$$

Brief Introduction

~~相信大家都会这个东西~~

不会的同学可以去问 zqc ~~小解杰~~

$$\max(S) = \sum_{T \in S, T \neq \varnothing} (-1)^{|T|+1} \min(T)$$

当然, min-max 容斥在期望意义下也满足

需要注意的是, 普通情况下期望的 $\min$ $\max$ 是比较难求的, 因为 $E[\max\{x, y\}] \neq \max\{E[x], E[y]\}$

Description

小 Z 有一个神奇的自动售货机，里面有 $n \times m$ 种物品，分别放在 n 行 m 列个格子中.

每当小 Z 向自动售货机中投入一枚硬币，他就能获得一对相邻格子中的物品 (已经获得的物品可能再次获得)，获得每一对相邻格子中的物品的概率是相等的.

在这 $n \times m$ 种物品中，有一些物品是小 Z 喜欢的，他想把这些物品包装成一份礼物.

小 Z 想知道，期望投入多少枚硬币后，就可以获得这些他喜欢的物品.

$n \le 6, m \le 100$

Solution

首先可以 **min-max 容斥**, 将所有物品被获得的时间的最大值的期望, 转化成关键点每个子集被获得时间的最小值的期望, 即

$$\max(S) = \sum_{T \in S, T \neq \varnothing} (-1)^{|T|+1} \min(T)$$

Solution

首先可以 **min-max 容斥**, 将所有物品被获得的时间的最大值的期望, 转化成关键点每个子集被获得时间的最小值的期望, 即

$$\max(S) = \sum_{T\in S, T\neq\varnothing} (-1)^{|T|+1} \min(T)$$

对于子集 T, 最早得到的物品的期望时间 $\min(T)$ 就是一次能选到子集中任意一个物品的概率的倒数

Solution

首先可以 **min-max 容斥**，将所有物品被获得的时间的最大值的期望，转化成关键点每个子集被获得时间的最小值的期望，即

$$\max(S) = \sum_{T \in S, T \neq \varnothing} (-1)^{|T|+1} \min(T)$$

对于子集 T，最早得到的物品的期望时间 $\min(T)$ 就是一次能选到子集中任意一个物品的概率的倒数

Proof.

设期望为 x，概率为 p，那么 $x = p + (1-p)(x+1)$，解得 $x = \frac{1}{p}$ □

Solution

首先可以 **min-max 容斥**, 将所有物品被获得的时间的最大值的期望, 转化成关键点每个子集被获得时间的最小值的期望, 即

$$\max(S) = \sum_{T \in S, T \neq \varnothing} (-1)^{|T|+1} \min(T)$$

对于子集 T, 最早得到的物品的期望时间 $\min(T)$ 就是一次能选到子集中任意一个物品的概率的倒数

Proof.

设期望为 x, 概率为 p, 那么 $x = p + (1-p)(x+1)$, 解得 $x = \frac{1}{p}$ □

而这个概率为 $\frac{cover(T)}{2nm-n-m}$, 其中 $cover(T)$ 表示有多少中可能的相邻格子对, 满足包含至少一个 T 中的物品

Solution

首先可以 **min-max 容斥**, 将所有物品被获得的时间的最大值的期望, 转化成关键点每个子集被获得时间的最小值的期望, 即

$$\max(S) = \sum_{T \in S, T \neq \varnothing} (-1)^{|T|+1} \min(T)$$

对于子集 T, 最早得到的物品的期望时间 $\min(T)$ 就是一次能选到子集中任意一个物品的概率的倒数

Proof.

设期望为 x, 概率为 p, 那么 $x = p + (1-p)(x+1)$, 解得 $x = \frac{1}{p}$ □

而这个概率为 $\frac{cover(T)}{2nm-n-m}$, 其中 $cover(T)$ 表示有多少中可能的相邻格子对, 满足包含至少一个 T 中的物品

考虑枚举 $cover(T)$, 通过 dp 求出每个 $cover(T)$ 的 $\sum(-1)^{|T|+1}$

Solution

如何计算才不会算重？

Solution

如何计算才不会算重？

对于一种格子对，我们只在靠上方/左边的格子处计算贡献，那么在 dp 的时候就需要知道它上方/左边的格子的状态 (是否在集合中)

Solution

如何计算才不会算重？

对于一种格子对，我们只在靠上方/左边的格子处计算贡献，那么在 dp 的时候就需要知道它上方/左边的格子的状态 (是否在集合中)

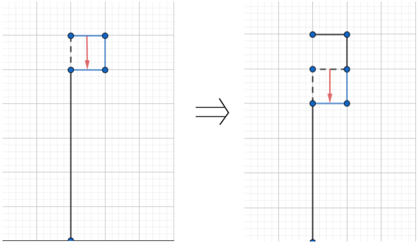
于是可以轮廓线 dp，状压的时候记录贴近轮廓线内一圈的状态，转移的时候推着轮廓线一格格走

Solution

如何计算才不会算重？

对于一种格子对，我们只在靠上方/左边的格子处计算贡献，那么在 dp 的时候就需要知道它上方/左边的格子的状态 (是否在集合中)

于是可以轮廓线 dp，状压的时候记录贴近轮廓线内一圈的状态，转移的时候推着轮廓线一格格走



这样转移即能知道当前左边和上方格子的信息

设 $dp[j][i][s][k]$ 表示当前轮廓线移动到第 i 行第 j 列这个格子, 轮廓线内状态为 s(是否被选入集合 T), 目前 $cover(T)=k$ 的 $\sum(-1)^{|T|+1}$

设 $dp[j][i][s][k]$ 表示当前轮廓线移动到第 i 行第 j 列这个格子, 轮廓线内状态为 s(是否被选入集合 T), 目前 $cover(T)=k$ 的 $\sum(-1)^{|T|+1}$

转移就是枚举当前这个格子选不选

设 $dp[j][i][s][k]$ 表示当前轮廓线移动到第 i 行第 j 列这个格子，轮廓线内状态为 s(是否被选入集合 T)，目前 $cover(T)=k$ 的 $\sum(-1)^{|T|+1}$

转移就是枚举当前这个格子选不选

显然前两维可以直接滚掉，因为每次只需从上一个状态转移过来

设 $dp[j][i][s][k]$ 表示当前轮廓线移动到第 i 行第 j 列这个格子，轮廓线内状态为 s(是否被选入集合 T)，目前 $cover(T)=k$ 的 $\sum(-1)^{|T|+1}$

转移就是枚举当前这个格子选不选

显然前两维可以直接滚掉，因为每次只需从上一个状态转移过来

复杂度 $O(2^n n^2 m^2)$

树上高消

Brief Introduction

与主元法类似地, 在树上 dp 状态间相互影响时, 把 dp 式子表示成只与 $fa[x]$ 有关的式子, 即 $dp[x] = A[x] \times dp[fa[x]] + B[x]$ 的形式

Brief Introduction

与主元法类似地, 在树上 dp 状态间相互影响时, 把 dp 式子表示成只与 $fa[x]$ 有关的式子, 即 $dp[x] = A[x] \times dp[fa[x]] + B[x]$ 的形式

根据某些特殊的等量关系求出某些特殊点的 dp 值, 进而推出所有点的 dp 值

Description

给定一棵 n 个结点的树，你从点 x 出发，每次等概率随机选择一条与所在点相邻的边走过去.

有 Q 次询问，每次询问给定一个集合 S，求如果从 x 出发一直随机游走，直到点集 S 中所有点都至少经过一次的话，期望游走几步.

特别地，点 x（即起点）视为一开始就被经过了一次.

答案对 998244353 取模.

$n \leq 18, Q \leq 5000$

Solution

首先可以 **min-max 容斥**, 转化为求第一次访问集合 S 中点的期望时间

Solution

首先可以 **min-max 容斥**, 转化为求第一次访问集合 S 中点的期望时间

先枚举集合 S, 然后树形 Dp, 设 $dp[x]$ 表示从 x 出发, 第一次访问 S 中节点的期望步数

Solution

首先可以 **min-max 容斥**, 转化为求第一次访问集合 S 中点的期望时间

先枚举集合 S, 然后树形 Dp, 设 $dp[x]$ 表示从 x 出发, 第一次访问 S 中节点的期望步数

- 若 $x \in S$: $dp[x] = 0$

Solution

首先可以 **min-max 容斥**, 转化为求第一次访问集合 S 中点的期望时间

先枚举集合 S, 然后树形 Dp, 设 $dp[x]$ 表示从 x 出发, 第一次访问 S 中节点的期望步数

- 若 $x \in S$: $dp[x] = 0$
- 若 $x \notin S$: 设 $d[x]$ 表示 x 的度数

$$
\begin{aligned}
dp[x] &= \frac{1}{d[x]}(dp[fa[x]] + 1 + \sum_{y} dp[y] + 1) \\
&= \frac{1}{d[x]} dp[fa[x]] + \frac{1}{d[x]}(\sum_{y} dp[y]) + 1
\end{aligned}
$$

Solution

首先可以 **min-max 容斥**, 转化为求第一次访问集合 S 中点的期望时间

先枚举集合 S, 然后树形 Dp, 设 $dp[x]$ 表示从 x 出发, 第一次访问 S 中节点的期望步数

- 若 $x \in S$: $dp[x] = 0$
- 若 $x \notin S$: 设 $d[x]$ 表示 x 的度数

$$\begin{aligned} dp[x] &= \frac{1}{d[x]}(dp[fa[x]] + 1 + \sum_y dp[y] + 1) \\ &= \frac{1}{d[x]} dp[fa[x]] + \frac{1}{d[x]}(\sum_y dp[y]) + 1 \end{aligned}$$

将 $dp[y]$ 表示成 $dp[y] = A[y] * dp[x] + B[y]$ 的形式

Solution

代入得到

$$dp[x] = \frac{1}{d[x]} dp[fa[x]] + \frac{1}{d[x]}(\sum_{y} A[y] * dp[x] + B[y]) + 1$$

Solution

代入得到

$$dp[x] = \frac{1}{d[x]} dp[fa[x]] + \frac{1}{d[x]} (\sum_y A[y] * dp[x] + B[y]) + 1$$

通过一些简单的推式子可以得到

$$\begin{cases} A[x] = \frac{1}{d[x] - \sum_y A[y]} \\ B[x] = \frac{(\sum_y B[y]) + d[x]}{d[x] - \sum_y A[y]} \end{cases}$$

然后就能不断往上 dp 了

Solution

代入得到

$$dp[x] = \frac{1}{d[x]} dp[fa[x]] + \frac{1}{d[x]} (\sum_y A[y] * dp[x] + B[y]) + 1$$

通过一些简单的推式子可以得到

$$\begin{cases} A[x] = \frac{1}{d[x] - \sum_y A[y]} \\ B[x] = \frac{(\sum_y B[y]) + d[x]}{d[x] - \sum_y A[y]} \end{cases}$$

然后就能不断往上 dp 了

需要注意的是, 计算答案时不能每次询问 $O(3^n)$ 枚举子集, 需要用高维前缀和 $O(n * 2^n)$ 预处理出每个集合的答案

贪心更新状态

Description

n 个点的完全图，每条边有 $p_{i,j}$ 的概率出现，求**最优策略下**从 1 号点到达 n 号点的期望距离

注：你的策略可以不断作出改变

$n \leq 2000$

Solution

贪心考虑不难发现，最优策略一定是每次尽量尝试走向期望时间小的点，如果不能走再考虑第二小的，以此类推，直到某个点的期望时间比它大，那么就不往那边走，直接留在原地

Solution

贪心考虑不难发现，最优策略一定是每次尽量尝试走向期望时间小的点，如果不能走再考虑第二小的，以此类推，直到某个点的期望时间比它大，那么就不往那边走，直接留在原地

设 $dp[x]$ 表示 x 号点走到 n 号点最优的期望时间，a_i 表示 $dp[x]$ 第 i 小的 x，那么有

$$dp[a_i] = 1 + \sum_{j=1}^{i} dp[a_j] \cdot p_{a_i,a_j} \cdot \prod_{k=1}^{j-1}(1 - p_{a_i,a_k})$$

$$= \frac{1 + \sum_{j=1}^{i-1} dp[a_j] \cdot p_{a_i,a_j} \cdot \prod_{k=1}^{j-1}(1 - p_{a_i,a_k})}{1 - \prod_{j=1}^{i-1}(1 - p_{a_i,a_j})}$$

Solution

贪心考虑不难发现，最优策略一定是每次尽量尝试走向期望时间小的点，如果不能走再考虑第二小的，以此类推，直到某个点的期望时间比它大，那么就不往那边走，直接留在原地

设 $dp[x]$ 表示 x 号点走到 n 号点最优的期望时间，a_i 表示 $dp[x]$ 第 i 小的 x，那么有

$$dp[a_i] = 1 + \sum_{j=1}^{i} dp[a_j] \cdot p_{a_i,a_j} \cdot \prod_{k=1}^{j-1}(1 - p_{a_i,a_k})$$

$$= \frac{1 + \sum_{j=1}^{i-1} dp[a_j] \cdot p_{a_i,a_j} \cdot \prod_{k=1}^{j-1}(1 - p_{a_i,a_k})}{1 - \prod_{j=1}^{i-1}(1 - p_{a_i,a_j})}$$

显然 $a_1 = n$，后面的 a_i 都可以通过不断找出当前局面最小的 dp 值求出

Solution

贪心考虑不难发现，最优策略一定是每次尽量尝试走向期望时间小的点，如果不能走再考虑第二小的，以此类推，直到某个点的期望时间比它大，那么就不往那边走，直接留在原地

设 $dp[x]$ 表示 x 号点走到 n 号点最优的期望时间，a_i 表示 $dp[x]$ 第 i 小的 x，那么有

$$dp[a_i] = 1 + \sum_{j=1}^{i} dp[a_j] \cdot p_{a_i,a_j} \cdot \prod_{k=1}^{j-1}(1 - p_{a_i,a_k})$$

$$= \frac{1 + \sum_{j=1}^{i-1} dp[a_j] \cdot p_{a_i,a_j} \cdot \prod_{k=1}^{j-1}(1 - p_{a_i,a_k})}{1 - \prod_{j=1}^{i-1}(1 - p_{a_i,a_j})}$$

显然 $a_1 = n$，后面的 a_i 都可以通过不断找出当前局面最小的 dp 值求出

分别维护一下分子部分以及 $\prod$ 的部分即可，时间复杂度 $O(n^2)$

Description

给出一个 n 个点 m 条边的无向图，现在从 1 号点出发到 n 号点

假设现在在 i 号点买票，买到的票通向的是 i 号点相邻点的随机一个，买到的票可以选择立即使用 (去往票通向的节点) 或者直接扔掉 (需要重新在该节点买票)

问**最优决策下**期望需要买多少张票能到 n 号点.

$n, m \leq 10^5$

Solution

设 $dp[x]$ 表示从 x 出发, 最优策略下走到 n 的期望买票数, 那么

$$dp[x] = 1 + \frac{1}{deg[x]} \sum_{y \in adj(x)} \min\{dp[x], dp[y]\}$$

Solution

设 $dp[x]$ 表示从 x 出发, 最优策略下走到 n 的期望买票数, 那么

$$dp[x] = 1 + \frac{1}{deg[x]} \sum_{y \in adj(x)} \min\{dp[x], dp[y]\}$$

假设 x 有 $cnt[x]$ 个相邻节点 y, 满足 $dp[y] < dp[x]$, 那么有

$$dp[x] = \frac{deg[x] + \sum_{y \in adj(x), dp[y] < dp[x]} dp[y]}{cnt[x]}$$

Solution

设 $dp[x]$ 表示从 x 出发, 最优策略下走到 n 的期望买票数, 那么

$$dp[x] = 1 + \frac{1}{deg[x]} \sum_{y \in adj(x)} \min\{dp[x], dp[y]\}$$

假设 x 有 $cnt[x]$ 个相邻节点 y, 满足 $dp[y] < dp[x]$, 那么有

$$dp[x] = \frac{deg[x] + \sum_{y \in adj(x), dp[y] < dp[x]} dp[y]}{cnt[x]}$$

一开始 $dp[n] = 0$, 其他 dp 值为 ∞,cnt 为 0

Solution

设 $dp[x]$ 表示从 x 出发，最优策略下走到 n 的期望买票数，那么

$$dp[x] = 1 + \frac{1}{deg[x]} \sum_{y \in adj(x)} \min\{dp[x], dp[y]\}$$

假设 x 有 $cnt[x]$ 个相邻节点 y，满足 $dp[y] < dp[x]$，那么有

$$dp[x] = \frac{deg[x] + \sum_{y \in adj(x), dp[y] < dp[x]} dp[y]}{cnt[x]}$$

一开始 $dp[n] = 0$，其他 dp 值为 ∞, cnt 为 0

每次找出当前最小的 $dp[x]$，再用 x 去更新周围的点，类似于最短路松弛的过程，复杂度 $O(m \log n)$

赌徒破产模型

Brief Introduction

一个赌徒一开始有 h 枚金币，每次有 p 的概率获得一枚金币或 $(1-p)$ 的概率丢掉一枚金币

若他的金币多于 T（赚够收手获胜）或者少于 S（破产）则游戏结束

求他在破产前（在多于 S 的前提下）获胜（达到 T）的概率

Brief Introduction

设 $f[i]$ 表示当前收益为 i 时刻的条件下能最终获胜的概率, 则显然有转移方程：

$$f[i] = p \cdot f[i+1] + (1-p) \cdot f[i-1]$$

Brief Introduction

设 $f[i]$ 表示当前收益为 i 时刻的条件下能最终获胜的概率，则显然有转移方程：

$$f[i]=p\cdot f[i+1]+(1-p)\cdot f[i-1]$$

这就是一个二阶常系数线性齐次递推式，我们知道它的特征方程:

$$x^2-\frac{1}{p}x+\frac{1-p}{p}=0$$

Brief Introduction

设 $f[i]$ 表示当前收益为 i 时刻的条件下能最终获胜的概率，则显然有转移方程：

$$f[i] = p \cdot f[i+1] + (1-p) \cdot f[i-1]$$

这就是一个二阶常系数线性齐次递推式，我们知道它的特征方程:

$$x^2 - \frac{1}{p}x + \frac{1-p}{p} = 0$$

它有两个特征根：$x_1 = 1, x_2 = \frac{1-p}{p}$

Brief Introduction

设 $f[i]$ 表示当前收益为 i 时刻的条件下能最终获胜的概率, 则显然有转移方程：

$$f[i] = p \cdot f[i+1] + (1-p) \cdot f[i-1]$$

这就是一个二阶常系数线性齐次递推式, 我们知道它的特征方程:

$$x^2 - \frac{1}{p}x + \frac{1-p}{p} = 0$$

它有两个特征根：$x_1 = 1, x_2 = \frac{1-p}{p}$, 于是得到通解：$f[n] = c_1(\frac{1-p}{p})^n + c_2$

然后再根据已知条件, 一般是与题中 S, T 相关的信息 (如 $f[S] = 0$, $f[T] = 1$ 之类的关系) 求出 c_1, c_2, 进而求出 $f[n]$

Description

N 只猫围成一圈玩游戏, 顺时针编号 0 到 $N-1$, $N-1$ 与 0 相邻. 游戏规则如下：

- 一开始编号 0 的猫拿着一个球
- 每个回合中手里拿球的猫抛硬币, 该硬币有 P 的概率正面朝上, $(1-P$ 的概率反面朝上
- 如果硬币正面朝上, 则该猫 j 把球传给右边 (编号为 $(j+1)\%N$) 的猫, 否则传给左边 (编号为 $(j-1+N)\%N$) 的猫
- 该游戏持续进行直到每只猫至少拿到一次球, 且最终拿球的猫赢得游戏

给定 N, K, P, 求编号为 K 的猫赢得游戏的概率

$N \leq 10^9$

Solution

通过观察发现，如果最后是猫 K 拿到球，那么上一轮一定是 $K-1$ 号猫或者 $K+1$ 号猫拿球，且除了 K 号猫之外所有的猫都拿过球

Solution

通过观察发现, 如果最后是猫 K 拿到球, 那么上一轮一定是 $K-1$ 号猫或者 $K+1$ 号猫拿球, 且除了 K 号猫之外所有的猫都拿过球

因此答案就是 $P(K+1|K-1)+P(K-1|K+1)$

Solution

通过观察发现, 如果最后是猫 K 拿到球, 那么上一轮一定是 $K-1$ 号猫或者 $K+1$ 号猫拿球, 且除了 K 号猫之外所有的猫都拿过球

因此答案就是 $P(K+1|K-1)+P(K-1|K+1)$

不难发现这实际上就是把模型的上下边界调整了一下 (把这个圈展开), 直接上模型即可

Description

袋子里有 n 种球, 第 i 种颜色有 a_i 个

每次操作随机选两个球, 将第一个球染成第二个球的颜色

求全部颜色变成相同的期望次数

$1 \leq n \leq 2500, 1 \leq a_i \leq 10^5$

Solution

先枚举最终的颜色是什么，设 $f[i]$ 表示当前有 i 个钦定颜色的球，到终止的期望次数，那么

$$f[i] = p(f[i-1] + f[i+1]) + (1-2p)f[i] + E$$

Solution

先枚举最终的颜色是什么，设 $f[i]$ 表示当前有 i 个钦定颜色的球，到终止的期望次数，那么

$$f[i] = p(f[i-1] + f[i+1]) + (1-2p)f[i] + E$$

记 $s = \sum_{i=1}^{n} a_i$，其中 $p = \frac{i(s-i)}{s(s-1)}$，为选出一个钦定的球和一个其他颜色球的概率，变多变少都是这个概率

Solution

先枚举最终的颜色是什么，设 $f[i]$ 表示当前有 i 个钦定颜色的球，到终止的期望次数，那么

$$f[i]=p(f[i-1]+f[i+1])+(1-2p)f[i]+E$$

记 $s=\sum_{i=1}^{n}a_i$，其中 $p=\frac{i(s-i)}{s(s-1)}$，为选出一个钦定的球和一个其他颜色球的概率，变多变少都是这个概率

而 $E=\frac{i}{s}$，为产生贡献的**期望次数**

Solution

先枚举最终的颜色是什么，设 $f[i]$ 表示当前有 i 个钦定颜色的球，到终止的期望次数，那么

$$f[i] = p(f[i-1] + f[i+1]) + (1-2p)f[i] + E$$

记 $s = \sum_{i=1}^{n} a_i$，其中 $p = \frac{i(s-i)}{s(s-1)}$，为选出一个钦定的球和一个其他颜色球的概率，变多变少都是这个概率

而 $E = \frac{i}{s}$，为产生贡献的**期望次数**

为什么这里 $E \neq 1$ 呢？

Solution

先枚举最终的颜色是什么，设 $f[i]$ 表示当前有 i 个钦定颜色的球，到终止的期望次数，那么

$$f[i] = p(f[i-1] + f[i+1]) + (1-2p)f[i] + E$$

记 $s = \sum_{i=1}^{n} a_i$，其中 $p = \frac{i(s-i)}{s(s-1)}$，为选出一个钦定的球和一个其他颜色球的概率，变多变少都是这个概率

而 $E = \frac{i}{s}$，为产生贡献的**期望次数**

为什么这里 $E \neq 1$ 呢？

因为当前钦定的颜色必须存在，个数不能为 0，即当前颜色球的个数要在到达 0 之前到达 s

Solution

先枚举最终的颜色是什么，设 $f[i]$ 表示当前有 i 个钦定颜色的球，到终止的期望次数，那么

$$f[i]=p(f[i-1]+f[i+1])+(1-2p)f[i]+E$$

记 $s=\sum_{i=1}^{n}a_i$，其中 $p=\frac{i(s-i)}{s(s-1)}$，为选出一个钦定的球和一个其他颜色球的概率，变多变少都是这个概率

而 $E=\frac{i}{s}$，为产生贡献的**期望次数**

为什么这里 $E\neq 1$ 呢？

因为当前钦定的颜色必须存在，个数不能为 0，即当前颜色球的个数要在到达 0 之前到达 s

这实际上就是一个赌徒破产模型，通过解方程能得到从 i 个球在到 0 个球之前到 s 个球的概率为 $\frac{i}{s}$

Solution

化简 dp 式得到：$2f[i] = f[i-1] + f[i+1] + \frac{s-1}{s-i}$

边界情况：$f[2] = 2f[1] - 1; f[s] = 0$

Solution

化简 dp 式得到：$2f[i] = f[i-1] + f[i+1] + \frac{s-1}{s-i}$

边界情况：$f[2] = 2f[1] - 1; f[s] = 0$

把 $f[1]$ 看作主元, 将 $f[i] - f[i+1]$ 用 $f[1]$ 表示：

$$f[i] - f[i+1] = f[i-1] - f[i] + \frac{s-1}{s-i} = -f[1] + \sum_{j=1}^{i} \frac{s-1}{s-j}$$

Solution

化简 dp 式得到：$2f[i] = f[i-1] + f[i+1] + \frac{s-1}{s-i}$

边界情况：$f[2] = 2f[1] - 1; f[s] = 0$

把 $f[1]$ 看作主元, 将 $f[i] - f[i+1]$ 用 $f[1]$ 表示：

$$f[i] - f[i+1] = f[i-1] - f[i] + \frac{s-1}{s-i} = -f[1] + \sum_{j=1}^{i} \frac{s-1}{s-j}$$

继续化简：

$$f[1] = \sum_{i=1}^{s-1} f[i] - f[i+1] = -(s-1)f[1] + \sum_{i=1}^{s-1} \frac{s-1}{s-i} \cdot (s-i)$$

Solution

化简 dp 式得到：$2f[i] = f[i-1] + f[i+1] + \frac{s-1}{s-i}$

边界情况：$f[2] = 2f[1] - 1; f[s] = 0$

把 $f[1]$ 看作主元, 将 $f[i] - f[i+1]$ 用 $f[1]$ 表示：

$$f[i] - f[i+1] = f[i-1] - f[i] + \frac{s-1}{s-i} = -f[1] + \sum_{j=1}^{i} \frac{s-1}{s-j}$$

继续化简：

$$f[1] = \sum_{i=1}^{s-1} f[i] - f[i+1] = -(s-1)f[1] + \sum_{i=1}^{s-1} \frac{s-1}{s-i} \cdot (s-i)$$

于是解出：$f[1] = \frac{(s-1)^2}{s}$

Solution

化简 dp 式得到：$2f[i]=f[i-1]+f[i+1]+\frac{s-1}{s-i}$

边界情况：$f[2]=2f[1]-1; f[s]=0$

把 $f[1]$ 看作主元, 将 $f[i]-f[i+1]$ 用 $f[1]$ 表示：

$$f[i]-f[i+1]=f[i-1]-f[i]+\frac{s-1}{s-i}=-f[1]+\sum_{j=1}^{i}\frac{s-1}{s-j}$$

继续化简：

$$f[1]=\sum_{i=1}^{s-1}f[i]-f[i+1]=-(s-1)f[1]+\sum_{i=1}^{s-1}\frac{s-1}{s-i}\cdot(s-i)$$

于是解出：$f[1]=\frac{(s-1)^2}{s}$

直接递推即能得到所有 f, 答案即为 $\sum_{i=1}^{n}f[a_i]$

The End

The End

Thanks

$P.S.$ 课件里有些题写得不太详细的

可以在我博客：hk-cnyali.com里找找更详细的题解