

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
●  
○  
○  
○  
○  
○  
○  
○  
○

MERMAID  
○  
○  
○  
○  
○  
○  
○

题目来源

# 题目来源

- NEERC2016 Regional D

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
●  
○  
○  
○  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

考察算法

# 考察算法

● 网络流

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
●  
○  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○

算法一

# 算法一

- $O(2^n)$ 暴力。

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
●  
○  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○

算法一

# 算法一

- $O(2^n)$ 暴力。

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
●  
○  
○  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法一

# 算法一

- $O(2^n)$ 暴力。
- 期望得分：1分

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
●  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○

算法二

# 算法二

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
●  
○  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法二

# 算法二

● 是时候展现真正的技术了!

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
●  
○  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法二

## 算法二

- 是时候展现真正的技术了!

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
●  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○

算法二

## 算法二

- 是时候展现真正的技术了!
- 爆搜一下，乱搞一下，退火一下，万一过了呢

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
●  
○  
○  
○○  
○○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法二

## 算法二

- 是时候展现真正的技术了!
- 爆搜一下，乱搞一下，退火一下，万一过了呢

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
●  
○  
○  
○○  
○○○

MERMAID

○  
○  
○  
○  
○  
○  
○

算法二

## 算法二

- 是时候展现真正的技术了!
- 爆搜一下，乱搞一下，退火一下，万一过了呢
- 期望得分：可能有 25 分

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
●  
○  
○○○

MERMAID  
○  
○  
○  
○  
○  
○

算法三

# 算法三

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
●  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法三

# 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
●  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○

算法三

## 算法三

- 设 $w_i$ 表示前 $i$ 天有多少个 $s$

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
●  
○  
○  
○

MERMAID

○  
○  
○  
○  
○  
○  
○

算法三

## 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$
- 对于所有  $i \geq k$  满足  $w_i - w_{i-k} \geq ms$  且  $w_i - w_{i-k} \leq k - me$

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
●  
○  
○  
○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法三

## 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$
- 对于所有  $i \geq k$  满足  $w_i - w_{i-k} \geq ms$  且  $w_i - w_{i-k} \leq k - me$

# 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$
- 对于所有  $i \geq k$  满足  $w_i - w_{i-k} \geq ms$  且  $w_i - w_{i-k} \leq k - me$
- 显然这个单纯形最优解一定是整数解

# 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$
- 对于所有  $i \geq k$  满足  $w_i - w_{i-k} \geq ms$  且  $w_i - w_{i-k} \leq k - me$
- 显然这个单纯形最优解一定是整数解

# 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$
- 对于所有  $i \geq k$  满足  $w_i - w_{i-k} \geq ms$  且  $w_i - w_{i-k} \leq k - me$
- 显然这个单纯形最优解一定是整数解
- **大力单纯形!**

# 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$
- 对于所有  $i \geq k$  满足  $w_i - w_{i-k} \geq ms$  且  $w_i - w_{i-k} \leq k - me$
- 显然这个单纯形最优解一定是整数解
- 大力单纯形!

# 算法三

- 设  $w_i$  表示前  $i$  天有多少个  $s$
- 对于所有  $i \geq k$  满足  $w_i - w_{i-k} \geq ms$  且  $w_i - w_{i-k} \leq k - me$
- 显然这个单纯形最优解一定是整数解
- 大力单纯形!
- 期望得分: 35分

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
●  
○○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法四

# 算法四

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
●  
○○○

MERMAID  
○  
○  
○  
○  
○  
○

算法四

## 算法四

- 观察算法三的单纯形，每一个约束都是 $X - Y \leq C$ 的形式

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
●  
○○○

MERMAID  
○  
○  
○  
○  
○  
○

算法四

## 算法四

- 观察算法三的单纯形，每一个约束都是 $X - Y \leq C$ 的形式

说明

○

ECHO

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

EXEC

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

MERMAID

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

○

算法四

## 算法四

- 观察算法三的单纯形，每一个约束都是 $X - Y \leq C$ 的形式
- 我们可以转对偶图后，跑最小费用可行流

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
○  
○  
●  
○  
○

MERMAID

○  
○  
○  
○  
○  
○  
○

算法四

## 算法四

- 观察算法三的单纯形，每一个约束都是 $X - Y \leq C$ 的形式
- 我们可以转对偶图后，跑最小费用可行流

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
○  
○  
●  
○○

MERMAID

○  
○  
○  
○  
○  
○

算法四

## 算法四

- 观察算法三的单纯形，每一个约束都是 $X - Y \leq C$ 的形式
- 我们可以转对偶图后，跑最小费用可行流
- 转对偶后的单纯形输方案有点麻烦，这里不讲了

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
○  
○  
●  
○○

MERMAID

○  
○  
○  
○  
○  
○

算法四

## 算法四

- 观察算法三的单纯形，每一个约束都是 $X - Y \leq C$ 的形式
- 我们可以转对偶图后，跑最小费用可行流
- 转对偶后的单纯形输方案有点麻烦，这里不讲了

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
○  
○  
●  
○○

MERMAID

○  
○  
○  
○  
○  
○

算法四

## 算法四

- 观察算法三的单纯形，每一个约束都是 $X - Y \leq C$ 的形式
- 我们可以转对偶图后，跑最小费用可行流
- 转对偶后的单纯形输方案有点麻烦，这里不讲了
- 期望得分：65 or 100分

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
●○○

MERMAID  
○  
○  
○  
○  
○  
○

算法五

# 算法五

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
●○○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

# 算法五

- 先假设每一天都是进行 $e$ 操作，这样就变成了把一些天变成 $s$ ，代价是 $s[i] - e[i]$





# 算法五

- 先假设每一天都是进行e操作，这样就变成了把一些天变成s，代价是 $s[i] - e[i]$
- 容易得到每k天变成s的个数必须在区间 $[ms, k - me]$ 内，在下文中我们假设他是 $[l, r]$

# 算法五

- 先假设每一天都是进行e操作，这样就变成了把一些天变成s，代价是 $s[i] - e[i]$
- 容易得到每k天变成s的个数必须在区间 $[ms, k - me]$ 内，在下文中我们假设他是 $[l, r]$
- 我们建立n个点，第i个点表示到第i天的信息。

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○●○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

# 建图

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○  
○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

## 建图

- $S$ 向 $1\dots k$ 所有点连一条 $\langle 1, 0 \rangle$ 的边

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○  
○

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

## 建图

- $S$ 向 $1\dots k$ 所有点连一条 $\langle 1, 0 \rangle$ 的边

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
○  
○  
○  
○  
○●○

MERMAID

○  
○  
○  
○  
○  
○  
○

算法五

## 建图

- $S$ 向 $1\dots k$ 所有点连一条 $\langle 1, 0 \rangle$ 的边
- $i$ 向 $i + 1$ 连一条 $\langle r - 1, 0 \rangle$ 的边

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
○  
○  
○  
○  
○●○

MERMAID

○  
○  
○  
○  
○  
○  
○

算法五

## 建图

- $S$ 向 $1\dots k$ 所有点连一条 $\langle 1, 0 \rangle$ 的边
- $i$ 向 $i + 1$ 连一条 $\langle r - 1, 0 \rangle$ 的边

说明

○

ECHO

○  
○  
○  
○  
○  
○  
○  
○  
○

EXEC

○  
○  
○  
○  
○  
○  
○  
○  
○  
○  
○

MERMAID

○  
○  
○  
○  
○  
○  
○  
○

算法五

## 建图

- $S$ 向 $1\dots k$ 所有点连一条 $\langle 1, 0 \rangle$ 的边
- $i$ 向 $i + 1$ 连一条 $\langle r - 1, 0 \rangle$ 的边
- $i$ 向 $i + k$ 连一条 $\langle 1, s[i] - e[i] \rangle$ 的边







## 建图

- $S$ 向 $1\dots k$ 所有点连一条 $\langle 1, 0 \rangle$ 的边
- $i$ 向 $i + 1$ 连一条 $\langle r - 1, 0 \rangle$ 的边
- $i$ 向 $i + k$ 连一条 $\langle 1, s[i] - e[i] \rangle$ 的边
- 以上的 $i + 1$ 和 $i + k$ ，如果大于 $n$ 就连到 $T$ 中
- 然后跑一发流量为 $r$ 的最大费用流就是答案

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○●

MERMAID  
○  
○  
○  
○  
○  
○

算法五

# 为什么呢？

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○●

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

## 为什么呢?

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○●

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○●

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$

说明  
○

ECHO  
○  
○  
○  
○  
○  
○  
○  
○

EXEC  
○  
○  
○  
○  
○  
○  
○  
○  
○●

MERMAID  
○  
○  
○  
○  
○  
○  
○

算法五

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$

## 为什么呢?

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$
- $i$ 向 $i+1$ 的边的流量只有 $r-1$ ，也就是说我们只能免费流 $r-1$ ，剩下的至少 $1$ 的流量必须在 $[i-k+1, i]$ 中流到 $i$ 后面

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$
- $i$ 向 $i+1$ 的边的流量只有 $r-1$ ，也就是说我们只能免费流 $r-1$ ，剩下的至少 $1$ 的流量必须在 $[i-k+1, i]$ 中流到 $i$ 后面

## 为什么呢?

- 可以发现,  $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程, 比如都流到了 $i$
- $i$ 向 $i+1$ 的边的流量只有 $r-1$ , 也就是说我们只能免费流 $r-1$ , 剩下的至少 $1$ 的流量必须在 $[i-k+1, i]$ 中流到 $i$ 后面
- 同样的, 因为流量是 $r$ , 所以 $[i-k+1, i]$ 中的流最多是 $r$

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$
- $i$ 向 $i+1$ 的边的流量只有 $r-1$ ，也就是说我们只能免费流 $r-1$ ，剩下的至少 $1$ 的流量必须在 $[i-k+1, i]$ 中流到 $i$ 后面
- 同样的，因为流量是 $r$ ，所以 $[i-k+1, i]$ 中的流最多是 $r$

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$
- $i$ 向 $i+1$ 的边的流量只有 $r-1$ ，也就是说我们只能免费流 $r-1$ ，剩下的至少 $1$ 的流量必须在 $[i-k+1, i]$ 中流到 $i$ 后面
- 同样的，因为流量是 $r$ ，所以 $[i-k+1, i]$ 中的流最多是 $r$
- 于是完美满足了题目中的条件

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$
- $i$ 向 $i+1$ 的边的流量只有 $r-1$ ，也就是说我们只能免费流 $r-1$ ，剩下的至少 $1$ 的流量必须在 $[i-k+1, i]$ 中流到 $i$ 后面
- 同样的，因为流量是 $r$ ，所以 $[i-k+1, i]$ 中的流最多是 $r$
- 于是完美满足了题目中的条件

## 为什么呢？

- 可以发现， $i$ 向 $i+k$ 连的边的意义就是第 $i$ 天是否变
- 我们脑补下流量的过程，比如都流到了 $i$
- $i$ 向 $i+1$ 的边的流量只有 $r-1$ ，也就是说我们只能免费流 $r-1$ ，剩下的至少 $1$ 的流量必须在 $[i-k+1, i]$ 中流到 $i$ 后面
- 同样的，因为流量是 $r$ ，所以 $[i-k+1, i]$ 中的流最多是 $r$
- 于是完美满足了题目中的条件
- 期望得分：100分