



قطار اسباب‌بازی

آرزو و برادرش برزو دوقلو هستند. آن‌ها یک مجموعه قطار اسباب‌بازی جالب برای تولدشان هدیه گرفته‌اند و می‌خواهند یک سیستم راه‌آهن با n ایستگاه و m خط آهن یک طرفه بسازند. ایستگاه‌ها از 0 تا $n - 1$ شماره‌گذاری شده‌اند. هر خط آهن از یک ایستگاه آغاز می‌شود و به همان ایستگاه یا یک ایستگاه متفاوت ختم می‌شود. از هر ایستگاه حداقل یک خط آهن آغاز می‌شود.

بعضی از ایستگاه‌ها، ایستگاه شارژکننده هستند. هنگامی که قطار به یک ایستگاه شارژکننده می‌رسد، به طور کامل شارژ می‌شود. قطار با شارژ کامل به اندازه کافی برای پیمودن n خط آهن متوالی شارژ دارد. یعنی پس از آن که شارژ شد، به محض ورود به $n + 1$ -امین خط آهن، شارژش تمام می‌شود.

در هر ایستگاه یک کلید وجود دارد که می‌تواند به هر کدام از خطوط آهنی که از آن ایستگاه آغاز می‌شوند، اشاره کند. هنگامی که یک قطار وارد یک ایستگاه می‌شود، از خط آهنی که کلید این ایستگاه به آن اشاره می‌کند خارج می‌شود.

دوقلوها می‌خواهند یک بازی با قطارشان انجام بدهند. آن‌ها همه ایستگاه‌ها را بین خودشان تقسیم کرده‌اند: هر ایستگاه یا متعلق به آرزوست و یا متعلق به برزو. تنها یک قطار وجود دارد. در آغاز بازی قطار در ایستگاه s قرار دارد و کاملاً شارژ شده است. برای شروع بازی، صاحب ایستگاه s کلید ایستگاه s را به یکی از خطوط آهنی که از ایستگاه s آغاز می‌شود، اشاره می‌دهد. سپس قطار را روشن می‌کنند و قطار حرکت خود را در خطوط آهن آغاز می‌کند.

هنگامی که یک قطار برای نخستین بار وارد یک ایستگاه می‌شود، صاحب آن ایستگاه کلید آن ایستگاه را تنظیم می‌کند. هنگامی که کلید تنظیم شود، وضعیت کلید، یعنی خط آهنی که کلید به آن اشاره می‌کند، تا آخر بازی ثابت می‌ماند. بنابراین اگر قطار دوباره وارد ایستگاهی شود که قبلاً آن را ملاقات کرده است، از همان خط آهنی که قبلاً خارج شده است، دوباره خارج خواهد شد.

از آنجایی که تعداد محدودی ایستگاه وجود دارد، قطار در نهایت وارد یک دور خواهد شد. یک دور دنباله‌ای از ایستگاه‌های متفاوت $c[0], c[1], \dots, c[k-1]$ است به طوری که قطار ایستگاه $c[i]$ (برای $0 \leq i < k - 1$) را با خط آهنی که به سمت ایستگاه $c[i+1]$ می‌رود ترک می‌کند، و ایستگاه $c[k-1]$ را با خط آهنی که به سمت ایستگاه $c[0]$ می‌رود ترک می‌کند. توجه کنید یک دور ممکن است تنها شامل یک ایستگاه (یعنی $k = 1$) باشد اگر قطار ایستگاه $c[0]$ را با استفاده از خط آهنی که به ایستگاه $c[0]$ بازمی‌گردد ترک کند.

آرزو برنده بازی خواهد بود اگر قطار به صورت نامتناهی به حرکت خود ادامه دهد، و برزو برنده بازی خواهد بود اگر شارژ قطار تمام شود. به عبارت دیگر، اگر حداقل یک ایستگاه شارژکننده در بین $c[0], c[1], \dots, c[k-1]$ وجود داشته باشد، قطار دوباره شارژ می‌شود و دور بدون انتها ادامه پیدا می‌کند، و آرزو برنده می‌شود. در غیر این صورت، شارژ قطار (شاید بعد از چندین بار چرخیدن در دور) بالاخره تمام می‌شود و برزو برنده می‌شود.

به شما توصیف سیستم خط آهن داده شده است. آرزو و برزو می‌خواهند n بازی انجام دهند. در بازی s -ام، برای $0 \leq s \leq n - 1$ ، قطار در ابتدا در ایستگاه s خواهد بود. شما باید برای هر بازی پیدا کنید که آیا یک استراتژی برای آرزو وجود دارد که برد آرزو را، علیرغم چگونگی بازی برزو، تضمین کند.

جزئیات پیاده‌سازی

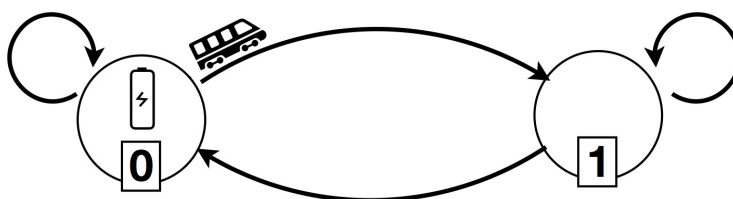
شما باید این تابع را پیاده‌سازی کنید:

```
int[] who_wins(int[] a, int[] r, int[] u, int[] v)
```

- a : آرایه‌ای به طول n . اگر آرزو صاحب ایستگاه i باشد، $a[i] = 1$. در غیر این صورت، برزو صاحب ایستگاه i است و $a[i] = 0$.
- r : آرایه‌ای به طول n . اگر ایستگاه i یک ایستگاه شارژکننده باشد، $r[i] = 1$. در غیر این صورت، $r[i] = 0$.
- u و v : آرایه‌هایی به طول m . برای هر $0 \leq i \leq m - 1$ ، یک خط آهن یک طرفه که از ایستگاه $u[i]$ آغاز می‌شود و در ایستگاه $v[i]$ خاتمه می‌یابد وجود دارد.
- این تابع باید آرایه w به طول n را بازگرداند. برای هر $0 \leq i \leq n - 1$ ، مقدار $w[i]$ باید 1 باشد اگر آرزو بتواند بازی که از ایستگاه i آغاز می‌شود را، علیرغم چگونگی بازی برزو، ببرد. در غیر این صورت، مقدار $w[i]$ باید 0 باشد.

مثال

```
who_wins([0, 1], [1, 0], [0, 0, 1, 1], [0, 1, 0, 1])
```



- دو ایستگاه وجود دارد. برزو صاحب ایستگاه 0 است که یک ایستگاه شارژکننده است. آرزو صاحب ایستگاه 1 است که ایستگاهی شارژکننده نیست.
- 4 خط آهن $(0, 0)$ ، $(0, 1)$ ، $(1, 0)$ ، و $(1, 1)$ وجود دارد که (i, j) یک خط آهن یک طرفه که از ایستگاه i آغاز می‌شود و به ایستگاه j خاتمه می‌یابد را نشان می‌دهد.
- بازی را در نظر بگیرید که در ابتدا قطار در ایستگاه 0 قرار دارد. اگر برزو کلید ایستگاه 0 را به سمت خط آهن $(0, 0)$ تنظیم کند، قطار به صورت نامتناهی در این خط آهن دور خواهد زد (توجه کنید ایستگاه 0 یک ایستگاه شارژکننده است). در این حالت آرزو برنده می‌شود. در غیر این صورت، اگر برزو کلید ایستگاه 0 را به سمت خط آهن $(0, 1)$ تنظیم کند، آرزو می‌تواند کلید ایستگاه 1 را به سمت خط آهن $(1, 0)$ تنظیم کند. در این صورت، قطار به صورت نامتناهی در بین دو ایستگاه دور خواهد زد، و از آنجایی که ایستگاه 0 یک ایستگاه شارژکننده است و قطار متوقف نمی‌شود، دوباره آرزو برنده می‌شود. بنابراین آرزو می‌تواند بازی را ببرد، علیرغم هر حرکتی که برزو انجام بدهد.
- با استدلالی مشابه، در حالتی که بازی از ایستگاه 1 آغاز می‌شود، آرزو می‌تواند، علیرغم هر بازی که برزو انجام دهد، برنده شود. بنابراین تابع باید $[1, 1]$ را برگرداند.

محدودیت‌ها

- $1 \leq n \leq 5000$
- $n \leq m \leq 20000$
- حداقل یک ایستگاه شارژکننده وجود دارد.

- از هر ایستگاه حداقل یک خط آهن خارج می‌شود.
- ممکن است برخی از خط‌های آهن از یک ایستگاه شروع شده و به همان ایستگاه ختم شوند (یعنی $u[i] = v[i]$).
- هر دو خط آهنی متفاوت هستند. به عبارت دیگر، هیچ دو اندیس i و j ($0 \leq i < j \leq m - 1$) وجود ندارند که برای آن‌ها داشته باشیم $u[i] = u[j]$ و $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n - 1$ (برای هر $0 \leq i \leq m - 1$).

زیرمسئله‌ها

1. (5 امتیاز) برای هر $0 \leq i \leq m - 1$ ، داریم $v[i] = u[i] + 1$ یا $v[i] = u[i]$.
2. (10 امتیاز) $n \leq 15$.
3. (11 امتیاز) آرزو صاحب همه ایستگاه‌هاست.
4. (11 امتیاز) برزو صاحب همه ایستگاه‌هاست.
5. (12 امتیاز) دقیقاً یک ایستگاه شارژکننده وجود دارد.
6. (51 امتیاز) بدون محدودیت.

ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

- سطر 1: n m
- سطر 2: $a[0]$ $a[1]$... $a[n - 1]$
- سطر 3: $r[0]$ $r[1]$... $r[n - 1]$
- سطر $4 + i$ (برای $0 \leq i \leq m - 1$): $u[i]$ $v[i]$

ارزیاب نمونه خروجی `who_wins` را در قالب زیر چاپ می‌کند:

- سطر 1: $w[0]$ $w[1]$... $w[n - 1]$