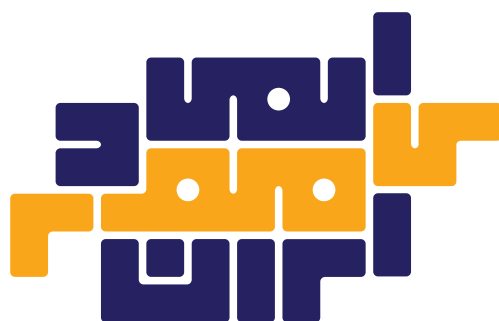


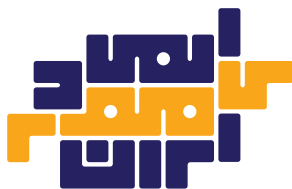


المپیاد کامپیوتر ایران

آزمون نهایی دوره ی تابستان سی و ششمین المپیاد کامپیوتر ایران

آزمون عملی اول - ۲۷ شهریور ۱۴۰۵

- آزمون شامل سه سوال است. زمان آزمون ۵ ساعت است.
- برای هر سوال می توانید حداکثر ۵۰ ارسال انجام دهید.
- بین هر دو ارسال متوالی برای یک سوال باید حداقل ۱ دقیقه فاصله باشد.
- موارد مبهم در سوالات را فقط از طریق قسمت Communication بپرسید.
- ترتیب سوالات به صورت کاملاً تصادفی انتخاب شده اند.
- نام کاربری شما برای ورود به سیستم day1 و رمز عبورتان arshia است.



کاشی‌های طلسم‌شده (runes)

در یک بازی پازل باستانی، نواری شامل $2n$ کاشی سنگی در یک ردیف چیده شده‌اند. روی این کاشی‌ها نمادهایی از ۱ تا n حک شده و از هر نماد دقیقاً دو کاشی در نوار وجود دارد. دنباله‌ی a به طول $2n$ وضعیت اولیه‌ی کاشی‌ها را از چپ به راست نشان می‌دهد.

بازیکن در ابتدای بازی می‌تواند یک کاشی کمکی با عدد $\circ$ را در یکی از $2n - 1$ فاصله‌ی میان دو کاشی مجاور قرار دهد. پس از قرارگیری این کاشی، می‌توان حرکت زیر را تا حداکثر n بار تکرار کرد:

دو کاشی هم‌نماد با اندیس‌های p و $p + 2$ را که دقیقاً یک کاشی میان آن‌ها فاصله انداخته است، انتخاب کرد ($a_p = a_{p+2}$). با لمس این دو کاشی، طلسم آن‌ها فعال شده و هر دو از نوار حذف می‌شوند.

پس از هر حذف، کاشی‌های باقی‌مانده با حفظ ترتیب اولیه‌شان در کنار یکدیگر قرار می‌گیرند. هدف بازی این است که با انجام n حرکت، تمامی $2n$ کاشی اصلی حذف شوند و در پایان تنها کاشی $\circ$ در ردیف باقی بماند. تعداد جایگاه‌هایی را بیابید که قرار دادن کاشی $\circ$ در آن‌ها، پیروزی در بازی و پاک‌سازی کامل نوار را ممکن می‌سازد.

ورودی

در خط اول ورودی عدد صحیح T ، تعداد سناریوها، آمده است.

توضیحات هر سناریو به صورت زیر است:

در خط اول هر سناریو عدد n آمده است که تعداد نمادهای مختلف را نشان می‌دهد.

در خط دوم هر سناریو $2n$ عدد صحیح $a_1, a_2, \dots, a_{2n}$ آمده است که نمادهای کاشی‌ها را به ترتیب از چپ به راست مشخص می‌کند.

 T n $a_1 \ a_2 \ \dots \ a_{2n}$

خروجی

به ازای هر سناریو، در یک خط تعداد جایگاه‌های مجاز برای قرار دادن کاشی کمکی را چاپ کنید.

محدودیت‌ها

- $1 \leq T \leq 200$
- $1 \leq n \leq 3 \times 10^5$
- مجموع n ‌ها در همه‌ی سناریوها از 3×10^5 بیشتر نیست.
- طول a برابر $2n$ است.
- $1 \leq a_i \leq n$
- هر عدد از ۱ تا n دقیقاً دو بار در a ظاهر می‌شود.

زیرمسئله‌ها

محدودیت‌ها	نمره	زیرمسئله
ورودی و خروجی نمونه	۰	۱
$n \leq 16$	۹	۲
$n \leq 70$	۱۰	۳
$n \leq 800$	۳۷	۴
بدون محدودیت اضافی	۴۴	۵

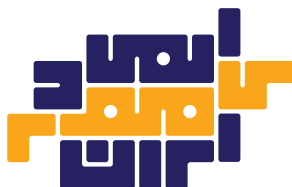
ورودی و خروجی نمونه

ورودی استاندارد	خروجی استاندارد
3	3
3	0
1 2 1 3 2 3	2
3	
1 2 3 1 2 3	
3	
1 3 2 1 2 3	

شرح ورودی و خروجی نمونه

در نمونه‌ی اول، با قرار دادن کاشی کمکی در فاصله‌های اول، سوم یا پنجم می‌توان بازی را برد. اگر کاشی در جایگاه اول گذاشته شود، می‌توان به ترتیب جفت‌های ۳، سپس ۲ و در آخر ۱ را حذف کرد. اگر در جایگاه سوم قرار گیرد، حذف کاشی‌ها به ترتیب ۱، سپس ۳ و در نهایت ۲ امکان‌پذیر است.

در نمونه‌ی دوم، هیچ جایگاهی برای کاشی کمکی منجر به پاک‌سازی نوار نمی‌شود.



پرتقال (orange)

پیرمرد مهربون پس از شکست‌های متوالی در المپیادهای نوین تصمیم گرفت به کشاورزی روی بیاورد. او یک درخت پرتقال n رأسی ریشه‌دار کاشته که ریشه‌ی آن رأس ۱ است. روی هر رأس این درخت یک پرتقال قرار دارد و میزان نارنجی بودن پرتقال رأس i با عدد صحیح a_i مشخص می‌شود.

پیرمرد مهربون قصد دارد پرتقال‌ها را در بسته‌های دوتایی بفروشد. او از دوره‌های المپیاد بازاریابی به یاد دارد که برای فروش بهتر، اختلاف میزان نارنجی بودن پرتقال‌های هر بسته باید دقیقاً برابر با ۱ باشد. به بیان دیگر، دو رأس u و v می‌توانند در یک بسته قرار بگیرند اگر و تنها اگر $|a_u - a_v| = 1$ باشد.

علاوه بر این، پیرمرد مهربون در المپیاد میوه‌شناسی یاد گرفته است که چگونه پرتقال‌ها را رنگ بزند؛ بنابراین او می‌تواند میزان نارنجی بودن هر پرتقال دلخواه را به اندازه‌ی یک عدد صحیح و مثبت افزایش دهد. با این حال، پیرمرد مهربون می‌داند که این کار اخلاقی نیست؛ از این رو می‌خواهد به ازای هر رأس v ، کمترین تعداد پرتقال‌هایی را که باید رنگ شوند تا بتوان تمام پرتقال‌های زیردرخت v را به‌طور کامل در بسته‌های معتبر قرار داد را محاسبه کند یا بفهمد که این کار ممکن نیست. متأسفانه پیرمرد مهربون در المپیاد کامپیوتر شرکت نکرده و اکنون از شما می‌خواهد که به او کمک کنید.

توجه کنید که پاسخ هر زیردرخت کاملاً مستقل از سایر زیردرخت‌ها محاسبه می‌شود؛ بنابراین رنگ زدن پرتقال‌ها برای یک زیردرخت، تأثیری بر بررسی سایر زیردرخت‌ها نخواهد داشت.

ورودی

در سطر اول، عدد صحیح n (تعداد رأس‌های درخت) می‌آید.

در سطر دوم، n عدد صحیح $a_1, a_2, \dots, a_n$ می‌آیند که میزان نارنجی بودن هر رأس را نشان می‌دهند.

در سطر سوم، $n - 1$ عدد صحیح $p_2, p_3, \dots, p_n$ می‌آیند که در آن p_i نشان‌دهنده‌ی پدر رأس i است.

 n $a_1 \ a_2 \ \dots \ a_n$ $p_2 \ p_3 \ \dots \ p_n$

خروجی

در یک سطر، به ازای هر رأس به ترتیب از ۱ تا n ، کمترین تعداد پرتقال‌هایی را چاپ کنید که باید رنگ شوند تا تمام پرتقال‌های زیردرخت آن رأس قابل بسته‌بندی باشند. در صورتی که تحت هیچ شرایطی امکان بسته‌بندی کامل وجود نداشت، عدد ۱- را چاپ کنید.

محدودیت‌ها

- $1 \leq n \leq 3 \times 10^5$
- $1 \leq a_i \leq 10^9$
- $1 \leq p_i \leq n$
- تضمین می‌شود ورودی داده شده یک درخت است.

زیرمسئله‌ها

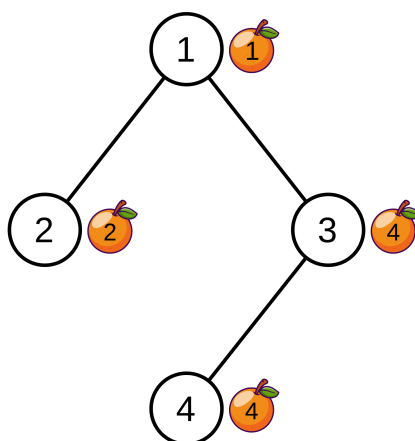
محدودیت‌ها	نمره	زیرمسئله
ورودی و خروجی نمونه	۵	۱
درخت یک مسیر با ریشه رأس ۱ است.	۲۱	۲
هیچ‌یک از اعداد a_i بر ۲ بخش پذیر نیست.	۴	۳
هیچ‌یک از اعداد a_i بر ۳ بخش پذیر نیست.	۸	۴
هیچ‌یک از اعداد a_i بر ۴ بخش پذیر نیست.	۱۰	۵
$n \leq 5000$	۹	۶
ارتفاع درخت (بیشترین فاصله از ریشه) حداکثر ۳۶ است.	۹	۷
بدون محدودیت اضافی	۳۹	۸

ورودی و خروجی نمونه

ورودی استاندارد	خروجی استاندارد
1 1	-1
4 1 2 4 4 1 1 3	1 -1 1 -1

شرح ورودی و خروجی نمونه

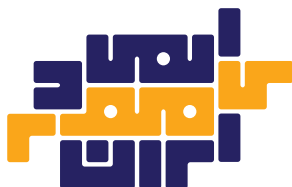
در نمونه‌ی دوم، درخت پرتقال به شکل زیر است. عدد داخل هر رأس شماره‌ی آن رأس و عدد داخل هر پرتقال میزان نارنجی بودن آن است.



برای زیردرخت رأس ۱، پیرمرد مهریون پرتقال رأس ۴ را رنگ می‌زند تا میزان نارنجی بودن آن به ۵ افزایش یابد؛ سپس پرتقال‌های رأس‌های ۱ و ۲ را در یک بسته و پرتقال‌های رأس‌های ۳ و ۴ را در بسته‌ای دیگر قرار می‌دهد.

برای زیردرخت رأس ۳، پیرمرد مهریون پرتقال رأس ۳ را رنگ می‌زند تا میزان نارنجی بودن آن به ۵ برسد؛ سپس پرتقال‌های رأس‌های ۳ و ۴ را در یک بسته قرار می‌دهد.

برای زیردرخت رأس‌های ۲ و ۴، چون تعداد رئوس فرد است (یک رأس)، امکان ایجاد بسته‌های دوتایی وجود ندارد و پاسخ ۱- است.



اتوبوس گردشگری (bus)

شبکه‌ی معابر یک شهرک شامل n ایستگاه است که با $n - 1$ جاده‌ی دوطرفه به یکدیگر متصل شده‌اند و تشکیل یک ساختار درختی می‌دهند که ریشه‌ی آن ایستگاه شماره‌ی ۱ است. پیمودن هر جاده در هر جهت دقیقاً ۱ واحد زمان طول می‌کشد. در هر ایستگاه v یک مسافر حضور دارد که مقصدش ایستگاه d_v است. این مسافران دارای محدودیت‌های زمانی هستند:

- مسافر ایستگاه v باید حداکثر در زمان a_v سوار اتوبوس شود.
- این مسافر باید حداکثر در زمان b_v در ایستگاه مقصد (d_v) پیاده شده باشد.

یک اتوبوس گردشگری با ظرفیت نامحدود در زمان ۰ از ایستگاه ۱ حرکت خود را آغاز می‌کند. اتوبوس در هر ایستگاه، فرزندان آن ایستگاه را به ترتیبی دلخواه برای پیمایش انتخاب می‌کند؛ به این صورت که وارد یک فرزند می‌شود، کل زیردرخت آن را به‌طور کامل می‌پیماید، سپس به این ایستگاه بازمی‌گردد و سراغ فرزند بعدی می‌رود. پس از پایان پیمایش تمامی فرزندان، اتوبوس به ایستگاه بالادست (پدر) بازمی‌گردد. با توجه به پیموده شدن هر جاده به‌صورت رفت و برگشت، حرکت اتوبوس دقیقاً در زمان $2n - 2$ و در ایستگاه ۱ به پایان می‌رسد.

سوار شدن مسافر هنگام حضور اتوبوس در مبدا و پیاده شدن او هنگام رسیدن اتوبوس به مقصد، به‌صورت آنی (بدون صرف زمان) انجام می‌شود. در صورتی که اتوبوس چند بار از یک ایستگاه عبور کند، مسافر می‌تواند در هر کدام از دفعات مجاز سوار شود و در یکی از دفعات بعدی حضور اتوبوس در مقصد، پیاده گردد.

مشخص کنید آیا می‌توان ترتیبی برای پیمایش فرزندان در هر ایستگاه انتخاب کرد که محدودیت‌های زمانی همه‌ی مسافران برآورده شود یا خیر.

ورودی

در سطر اول، عدد صحیح T ، تعداد سناریوها، آمده است. سپس توصیف سناریوها به‌ترتیب می‌آید.

در سطر اول هر سناریو، عدد صحیح n ، تعداد ایستگاه‌ها، آمده است.

در هر یک از $n - 1$ سطر بعدی، دو عدد صحیح u و v آمده است که نشان می‌دهند بین ایستگاه‌های u و v یک جاده وجود دارد.

در هر یک از n سطر بعدی، در سطر i ام، سه عدد صحیح d_i ، a_i و b_i برای مسافر ایستگاه i آمده است.

T
 n
 $u_1 \ v_1$
 $\dots$
 $u_{n-1} \ v_{n-1}$
 $d_1 \ a_1 \ b_1$
 $\dots$
 $d_n \ a_n \ b_n$

خروجی

برای هر سناریو، اگر انتخاب ترتیبی سازگار با شرایط همه‌ی مسافران ممکن است، عبارت YES را چاپ کنید. در غیر این صورت، عبارت NO را چاپ کنید.

محدودیت‌ها

- $1 \leq t \leq 3 \times 10^5$
- $1 \leq n \leq 3 \times 10^5$
- $1 \leq d_v \leq n$
- $0 \leq a_v, b_v \leq 2n - 2$
- جاده‌های داده‌شده در هر سناریو یک درخت معتبر می‌سازند.
- مجموع n در تمامی سناریوها حداکثر 3×10^5 است.

زیرمسئله‌ها

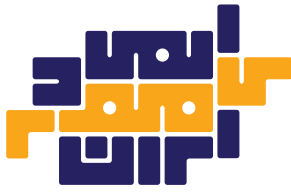
محدودیت‌ها	نمره	زیرمسئله
ورودی و خروجی نمونه	۰	۱
برای هر رأس $v > ۱$ داریم $p_v = v - ۱$.	۳	۲
برای هر رأس $v > ۱$ داریم $p_v = ۱$ و برای همه‌ی رئوس $a_v = b_v = ۲n - ۲$.	۷	۳
برای هر رأس $v > ۱$ داریم $p_v = ۱$.	۹	۴
برای همه‌ی رئوس $a_v = b_v = ۲n - ۲$.	۸	۵
برای همه‌ی رئوس $b_v = ۲n - ۲$.	۸	۶
برای همه‌ی رئوس $d_v = ۱$.	۶	۷
برای هر رأس v ، رأس d_v جد v است.	۱۱	۸
برای هر رأس v ، رأس d_v جد v نیست.	۲۱	۹
بدون محدودیت اضافی	۲۷	۱۰

ورودی و خروجی نمونه

ورودی استاندارد	خروجی استاندارد
2 3 1 3 1 2 1 0 0 3 1 3 1 3 4 3 1 2 1 3 1 0 0 3 3 4 2 3 4	YES NO

شرح ورودی و خروجی نمونه

در سناریوی اول نمونه، اتوبوس در ایستگاه ۱ ابتدا فرزند ۲ و سپس فرزند ۳ را می‌پیماید. در این صورت، دنباله‌ی عبور از ایستگاه‌ها به صورت $۱ \rightarrow ۳ \rightarrow ۱ \rightarrow ۲ \rightarrow ۱$ خواهد بود. مسافر ایستگاه ۲ در زمان ۱ سوار و در زمان ۳ در ایستگاه ۳ پیاده می‌شود. مسافر ایستگاه ۳ نیز در زمان ۳ سوار و در زمان ۴ در ایستگاه ۱ پیاده می‌شود.



کار با ارزیاب (grader)

در سؤالاتی که به کمک ارزیاب (Grader) داوری می‌شوند، لازم است ابتدا فایل‌های مورد نیاز را از صفحه‌ی آزمون دانلود کنید، برای این کار، می‌توانید به قسمت Attachments سوال مراجعه کنید.

پس از دانلود فایل‌های لازم و باز کردن آن‌ها، به پوشه‌ی cpp مراجعه کنید، در این پوشه معمولاً ۴ فایل زیر وجود دارد:

cpp

```
compile_cpp.sh  name.cpp  name.h  grader.cpp
```

که name نام سوال است.

پس از این لازم است کد توابعی که در صورت سوال ذکر شده‌اند را در فایل name.cpp پیاده‌سازی کنید، دقت کنید که حتماً به موارد زیر در پیاده‌سازی خود توجه کنید:

- خط ابتدایی فایل یعنی "#include \"name.h\" را حذف نکنید.
- نام توابع و پارامترهای آن‌ها را تغییر ندهید.
- هیچ چیزی را خروجی ندهید و چیزی را ورودی نگیرید، همچنین به هیچ وجه از تابع exit استفاده نکنید.
- توابعی به نام main در این فایل وجود ندارد و نباید اضافه کنید، در صورت اضافه کردن این تابع اجرا نخواهد شد.

پس از این، لازم است کد خود را به کمک اسکریپت آماده compile_cpp.sh کامپایل کنید، برای این کار، در همان پوشه cpp ترمینال را باز کنید (اگر موفق به باز کردن ترمینال در این پوشه نشدید، می‌توانید با کلیک راست روی این پوشه و انتخاب گزینه Open in Terminal این کار را انجام دهید) و دستور زیر را وارد کنید:

bash

```
bash compile_cpp.sh
```

پس از این، فایل اجرایی به نام name در همان پوشه ساخته خواهد شد، دوباره با اجرای دستور زیر در ترمینال، می‌توانید فایل ساخته شده توسط اسکریپت را اجرا کنید:

bash

```
./name
```

حال برای دیدن نحوه کار با ارزیاب نمونه و فرمت ورودی و خروجی آن، به قسمت ارزیاب نمونه صورت سوال مراجعه کنید. در نهایت پس از اطمینان از صحت عملکرد کد خود، می‌توانید فایل `name.cpp` را در صفحه آزمون بارگذاری کنید و نمره نهایی خود را دریافت کنید.

دقت کنید که لزومی ندارد که ارزیاب نمونه با ارزیاب داوری یکسان باشد، تنها تضمین این است که در صورتی که خواسته سوال را به درستی برآورده کرده باشید، ارزیاب داوری نیز نمره کامل را به شما خواهد داد.

برای دیباگ کد خود، می‌توانید محتویات فایل `grader.cpp` یا همان ارزیاب نمونه را تغییر دهید و یا حتی یک ارزیاب جدید بنویسید، اما توصیه می‌شود که زمان زیادی را صرف این کار نکنید.

در برخی از سوالات، به خصوص در سوالات اینترکتیو، ممکن است در سوال توابعی وجود داشته باشد که جزو توابعی که شما باید پیاده‌سازی کنید نیستند اما در صورت سوال وجود آن‌ها ذکر شده، در این صورت، شما می‌توانید فرض کنید که چه در ارزیاب نمونه و چه در ارزیاب داوری، یک پیاده‌سازی درست از این توابع وجود دارد و شما نیازی به پیاده‌سازی آن‌ها ندارید، و تنها کافی است تا از آن‌ها مستقیماً در پیاده‌سازی خودتان استفاده کنید.