انتخابی تیم - سی و یکمین دوره المپیاد کامپیوتر
۲۸ اردیبهشت ۱۴۰۱
آزمون عملی سوم

هیلفورد (Hillford)

Hillford

محدودیت زمانی: ۲ ثانیه
محدودیت حافظه: ۲۵۶ مگابایت

هیلفورد شهری کهن است که در نزدیکی چوسان قدیم واقع شده است. هیلفورد به دلیل قرار گرفتن در ارتفاعات دارای n تپه مختلف است. بین بعضی از این تپه‌ها جاده‌ی خاکی وجود دارد. طبق اطلاعات به جا مانده و نقشه‌های کهن به دست آمده، می‌دانیم گراف حاصل از تپه‌ها و جاده‌ها به شکل یک درخت است.

بعد از برپایی امپراطوری گوگوریو، فرمانده جومونگ که تپه‌ی فتح نشده باقی نگذاشته بود، تصمیم گرفت که هیلفورد را اشغال کرده و جهت آباد کردن آنجا تعدادی جاده‌ی آسفالت احداث کند!

استاد موپالمو به فرمانده پیشنهاد می‌دهد که بین دو تپه جاده آسفالت احداث کند اگر و تنها اگر جمع ارتفاع آن‌ها از تعداد جاده‌های خاکی مسیر بین‌شان بیشتر باشد. فرمانده جومونگ می‌داند که اگر به گفته استاد موپالمو گوش ندهد، استاد انقدر گریه می‌کند که جان به جان آفرین تسلیم می‌کند.

فرمانده جومونگ به حرف استاد موپالمو گوش می‌دهد و بین هر دو تپه v و u که $h_v + h_u \leq dis(v,u)$ می‌باشد، جاده آسفالت احداث می‌کند. $dis(v,u)$ تعداد جاده‌های خاکی بین دو تپه v و u است.

پس از احداث جاده‌ها، فرمانده جومونگ می‌خواهد تعدادی از افرادش را روی تپه‌ها گذاشته تا بتوانند از هیلفورد دفاع کنند. متاسفانه افراد جومونگ سر تقسیم غنائم جنگی با هم به مشکل خورده‌اند و دوست ندارند تپه هایی که بر روی آن قرار می‌گیرند، به یک دیگر توسط جاده آسفالت به طور مستقیم متصل باشد. فرمانده جومونگ برای امنیت هیلفورد می‌خواهد بداند که حداکثر چند نفر از افراد خود را می‌تواند روی تپه‌ها قرار دهد.

ورودی

در خط اول ورودی عدد n می‌آید که تعداد تپه‌های هیلفورد است.

در خط دوم ورودی n عدد می‌آید که ارتفاع تپه‌ها را نشان می‌دهد. iامین عدد ارتفاع تپه‌ی iام را نشان می‌دهد.

در هر یک از $n-1$ خط بعدی دو عدد u و v می‌آید که به معنای وجود جاده‌ی خاکی بین تپه u و تپه v است.

خروجی

در خروجی تنها یک عدد چاپ کنید که برابر حداکثر تعداد افرادی است که می‌توانند در هیلفورد مستقر شوند.

محدودیت‌ها

- $1 \leq n \leq 200000$
- $1 \leq h_i \leq n$

زیرمسئله‌ها

زیرمسئله	نمره	محدودیت‌ها
۱	۸	$۱ \leq n \leq ۲۰$
۲	۳۹	$۱ \leq n \leq ۲۰۰۰$
۳	۵۳	بدون محدودیت اضافی

ورودی و خروجی نمونه

خروجی استاندارد	ورودی استاندارد
1	2 1 2 1 2
3	8 2 1 2 1 2 2 1 1 2 1 8 3 3 2 5 6 7 8 6 4 7 5

انتخابی تیم - سی و یکمین دوره المپیاد کامپیوتر
۲۸ اردیبهشت ۱۴۰۱
آزمون عملی سوم

شعبده‌بازی کمینه بیشینه (MMQ)

Min Max Query

محدودیت زمانی: ۲ ثانیه
محدودیت حافظه: ۲۵۶ مگابایت

علی و رضا تصمیم گرفته‌اند که شعبده‌بازی کنند. شعبده‌بازی آن‌ها به این صورت است که شخصی از یک تماشاگر یک جایگشت از اعداد ۰ تا $n-1$ را روی کاغذ به علی می‌دهد تا رضا آن را نبیند. علی خوب به دنباله نگاه می‌کند و به نحوی که تماشاگران متوجه نشوند، با باز و بسته کردن چشم‌های خود یک دنباله باینری به طول k را به رضا می‌رساند. سپس علی از جمع می‌رود و رضا را با تماشاگران تنها می‌گذارد.

رضا از تماشاگران می‌خواهد تا از او درباره جایگشت انتخاب شده سوال بپرسند. سوالات تماشاگران به این صورت است که از رضا می‌پرسند که در دنباله‌ای که علی به او داده است، بیشینه یا کمینه بازه‌ی l تا r در کدام خانه قرار گرفته است. این یعنی رضا برای پاسخ عددی مثل x را می‌گوید که $l \leq x \leq r$ و با توجه به سوال تماشاگر، بیشینه یا کمینه این بازه است.

هر چه k کمتر باشد، علی کار ساده‌تری برای انجام دارد و چشم‌هایش را کمتر باز و بسته می‌کند. بنابراین علی و رضا می‌خواهند که k مقدار کمی داشته باشد. آن‌ها همیشه با $k \leq 3n$ می‌توانند به بهترین شکل شعبده‌بازی را اجرا کنند، ولی با مقادیری که در ادامه می‌بینید نیز می‌توانند شعبده‌بازی را با کیفیت کمتری اجرا کنند.

پیاده‌سازی

در این سوال شما اجازه خواندن از ورودی یا نوشتن در خروجی را ندارید و تنها باید توابعی که در ادامه گفته می‌شود را پیاده سازی کنید. در صورت هرگونه تلاش برای خواندن از ورودی یا نوشتن در خروجی یا استفاده از تابع exit نمره شما صفر می‌شود.

■ `void Ali(int n, int p[])`

این تابع را شما باید پیاده‌سازی کنید.

در ابتدا این تابع دقیقا یک فراخوانی می‌شود. در این تابع شما باید نقش علی را بازی کنید و باید بیت‌هایی که برای رضا می‌فرستید را تعیین کنید. برای ارسال بیت‌ها باید از تابع Send که در ادامه توضیح داده شده است استفاده کنید.

- n: طول جایگشتی که تماشاگر به علی داده است.
- p: جایگشتی که تماشاگر به علی داده است.

■ `void Reza(int n, int k, int A[])`

این تابع را شما باید پیاده‌سازی کنید.

بعد از فراخوانی تابع Ali این تابع دقیقا یک بار فراخوانی می‌شود. در این تابع به شما طول جایگشت اصلی، تعداد بیت‌هایی که علی برای شما فرستاده است و بیت‌هایی که علی برای شما فرستاده است داده می‌شود. در این تابع شما می‌تواند پردازش‌های مورد نیاز رضا را انجام دهید تا آماده پاسخ به سوالات تماشاگران شوید.

توجه کنید که پیش از فراخوانی این تابع تمامی متغیرهای برنامه شما به حالت اولیه برگردانده می‌شود. در نتیجه مقادیر متغیرهای global شما پس از اجرای تابع Ali وجود ندارند.

- n: طول جایگشتی که تماشاگر به علی داده است.
- k: تعداد بیت‌هایی که علی برای رضا فرستاده است.
- A: بیت‌هایی که علی برای رضا فرستاده است. $A[i]$ برابر با ۱ + iامین بیتی است که علی برای رضا فرستاده است.

■ `int Query(int t, int L, int R)`

این تابع را شما باید پیاده‌سازی کنید.

بعد از فراخوانی تابع Reza این تابع چندین بار فراخوانی می‌شود. بعد از پیش‌پردازش‌های لازم در تابع Reza تماشاگران از شما سوالاتی می‌پرسند. در این تابع شما باید به تماشاگران پاسخ دهید.

توجه کنید که شماره‌گذاری خانه‌های جایگشت از ۰ تا $n-1$ است؛ در نتیجه L و R بین ۰ و $n-1$ هستند. هم‌چنین هر دو خانه L و R در بازه‌ی پرسیده شده حضور دارند.

- t: اگر t برابر با ۱ باشد، تماشاگر می‌خواهد بداند که کمینه بازه‌ی L تا R کجاست. اگر t برابر با ۲ باشد، تماشاگر می‌خواهد بداند که بیشینه بازه‌ی L تا R کجاست.
- L: شروع بازه.
- R: پایان بازه.

■ `void Send(int bit)`

این تابع پیاده‌سازی شده است و شما تنها می‌توانید از آن استفاده کنید.

این تابع برای ارسال بیت‌ها در تابع Ali به رضا استفاده می‌شود.

- bit: بیتی که باید برای رضا فرستاده شود.

محدودیت‌ها

- $۱ \le n \le ۵۰۰۰$
- $۱ \le q \le ۳۰۰۰۰۰$

زیرمسئله‌ها

در صورتی که رضا حتی یک جواب اشتباه بدهد، نمره‌ی ارسال شما صفر خواهد بود. در غیر این صورت، فرض کنید شما k بار از تابع Send استفاده کرده‌اید؛ نمره‌ی شما به صورت زیر محاسبه می‌شود.

زیرمسئله	نمره	محدودیت‌ها
۱	۷	$k \le ۱۰^۵$
۲	$۷ + ۲۸ * \frac{\lceil log_۲(n) \rceil - \frac{k}{n}}{\lceil log_۲(n) \rceil - ۴}$	$۴n < k \le n * \lceil log_۲(n) \rceil$
۳	$۳۵ + ۶۵ * (۴ - \frac{k}{n})^۲$	$۳n < k \le ۴n$
۴	۱۰۰	$k \le ۳n$

در نهایت نمره شما از سوال کمترین نمره‌ای است که از هر کدام از تست‌ها می‌گیرید.

ارزیاب نمونه

فایل‌های مربوط به سوال را دانلود کنید. راه حل خود را درون `mmq.cpp` پیاده‌سازی کنید.
برای کامپایل و اجرا کردن برنامه باید `compile_cpp.sh` را اجرا کنید.
سپس فایل اجرایی `mmq` را که توسط `compile_cpp.sh` تولید شده است اجرا کنید.

ارزیاب نمونه ابتدا دو عدد n و q را می‌خواند که به‌ترتیب تعداد اعداد دنباله‌ی علی و تعداد سوال‌های تماشاگران است.
در خط بعدی n عدد می‌آید که دنباله‌ای است که تمشاگر به علی می‌گوید.
در q خط بعدی، در هر خط سه عدد می‌آیند که نشان دهنده‌ی t_i, L_i, R_i هستند.

ورودی و خروجی نمونه

ورودی استاندارد	خروجی استاندارد
6 5 2 3 1 5 0 4 1 1 3 1 2 5 2 0 2 1 3 5 2 1 4	2 4 1 4 3
10 10 9 0 4 3 8 2 6 7 1 5 1 1 9 2 5 8 1 6 7 2 2 5 1 3 4 2 7 8 1 8 9 2 2 7 1 0 9 2 0 9	1 7 6 4 3 7 8 4 1 0

انتخابی تیم - سی و یکمین دوره المپیاد کامپیوتر
۲۸ اردیبهشت ۱۴۰۱
آزمون عملی سوم

ربات‌ها (Robots)

robots

محدودیت زمانی: ۲ ثانیه
محدودیت حافظه: ۲۵۶ مگابایت

در جهانی موازی، قیصر امین‌پور و مولانا در مسابقه ربوکاپ شرکت کرده‌اند و هر یک رباتی ساخته‌اند. در ربوکاپ n ساختمان را پشت سر یکدیگر قرار می‌دهند که هر کدام ارتفاعی دارند. ارتفاع ساختمان iام از چپ h_i است. مسئولین ربوکاپ ربات قیصر و مولانا را روی ساختمان sام می‌گذارند. ربات‌ها می‌توانند بین ساختمان‌های متوالی جابه‌جا شوند؛ یعنی اگر ربات روی ساختمان iام باشد، می‌تواند به ساختمان $i+1$ و $i-1$ برود. ربات قیصر فقط به سمت چپ و ربات مولانا فقط به سمت راست حرکت می‌کند.

قیصر بر این باور است که «راهِ بالا رفتن مشکل و پیچان است، و سرازیر شدن مثل آب آسان است» و رباتش را به گونه‌ای ساخته که روی ساختمانی بلندتر از ساختمان اولیه‌اش نمی‌رود. به بیانی دیگر، ربات قیصر روی ساختمان s کارش را به پایان می‌رساند اگر هر دو شرط زیر برقرار باشد.

- $l = 1$ یا $h_{l-1} > h_s$.
- $h_j \leq h_s$ به ازای تمامی $l \leq j \leq s$ برقرار باشد.

مولانا بر خلاف شعر «ما ز بالاییم و بالا می‌رویم، ما ز دریاییم و دریا می‌رویم» رباتش را ساخته و رباتش فقط روی ساختمان‌هایی می‌رود که از ساختمان اولیه‌ایش کوتاه‌تر باشند. به بیانی دیگر، ربات مولانا روی ساختمان s کارش را به پایان می‌رساند اگر هر دو شرط زیر برقرار باشد.

- $r = n$ یا $h_{r+1} \geq h_s$.
- $h_j < h_s$ به ازای تمامی $s < j \leq r$ برقرار باشد.

مسئولین ربوکاپ به دنبال این هستند که مسابقه‌ی بین قیصر و مولانا جذاب شود. آن‌ها می‌توانند ارتفاع ساختمان iام را عددی طبیعی بین A_i تا B_i انتخاب کنند. از نظر آن‌ها مسابقه جذاب است اگر به ازای تمامی حالت‌های مختلف s اختلاف مسافتی که دو ربات طی می‌کنند از ۲ بیشتر نباشد. به عبارت دیگر، $|(s-l)-(r-s)| \leq 2$ به ازای تمامی s ($1 \leq s \leq n$) برقرار باشد. به چند طریق می‌توانند ارتفاع ساختمان‌ها را مشخص کنند تا مسابقه جذاب باشد؟

ورودی

در خط اول ورودی عدد طبیعی n می‌آید.

در هر یک از n خط بعدی، دو عدد طبیعی A_i و B_i به‌ترتیب می‌آیند.

خروجی

باقی‌مانده تقسیم جواب بر $10^9 + 7$ را پیدا کنید.

محدودیت‌ها

- $۱ \le n \le ۳۰۰$
- $۱ \le A_i \le B_i \le ۱۰^۹$

زیرمسئله‌ها

زیرمسئله	نمره	محدودیت‌ها
۱	۶	$B_i \le ۷$
۲	۷	$B_i \le ۱۰۰$
۳	۱۲	$B_i \le ۱۰۰۰۰$
۴	۱۱	$A_۱ = ۱, B_i = ۱۰^۹$
۵	۱۴	$n \le ۵۰$
۶	۵۰	بدون محدودیت اضافی

ورودی و خروجی نمونه

ورودی استاندارد	خروجی استاندارد
5 3 3 2 2 3 4 2 2 3 3	1

شرح ورودی و خروجی نمونه

در این نمونه مسئولین ربوکاپ دو حالت مختلف برای ارتفاع ساختمان‌ها می‌توانند انتخاب کنند.

- اگر ارتفاع ساختمان‌ها به ترتیب $\langle ۳, ۲, ۴, ۲, ۳ \rangle$ باشد: اگر ربات‌ها روی هر ساختمانی شروع کنند، اختلاف مسافتی که طی می‌کنند حداکثر ۲ است. در نتیجه در این صورت مسابقه جذاب است.
- اگر ارتفاع ساختمان‌ها به ترتیب $۳, ۲, ۳, ۲, ۳$ باشد. اگر $s = ۵$ باشد، ربات قیصر مسافت ۴ و ربات مولانا مسافت ۰ طی می‌کنند که اختلاف آن‌ها بیشتر از ۲ است. پس در این صورت مسابقه جذاب نیست.

با این حساب تنها یک انتخاب برای ارتفاع ساختمان‌ها وجود دارد که مسابقه جذاب باشد.