



مرتب‌سازی خطرناک (Sort)

زمان مجاز برای هر مورد: ۴۰۰ میلی‌ثانیه^۱
حافظه‌ی مصرفی مجاز: ۳۱ مگابایت^۲

در شرکت هیولاها، n هیولا مشغول به کار هستند. کمیته بازیابی اطلاعات قصد دارد هیولاها را بر حسب میزان حقوقی که می‌گیرند مرتب کند. مشکل نحوه‌ی صحبت هیولاها با آدم‌ها است. ما نمی‌توانیم مستقیماً از هیولاها مقدار حقوقشان را بپرسیم. تنها سئوالی که می‌توانیم بپرسیم این است: «کدام یک از دو هیولای i و j بیشتر حقوق می‌گیرد؟» می‌دانیم هیچ دو هیولایی مقدار حقوق برابر ندارند. به کمیته بازیابی اطلاعات کمک کنید تا با تعداد کمی پرسش، هیولاها را بر حسب حقوق مرتب کند. شما می‌توانید حداکثر ۲۰۰۰۰۰۰۰ سوال از هیولاها بپرسید و گرنه هیولاها عصبانی می‌شوند.

مسئله

برنامه‌ای بنویسید که

- تعداد هیولاها (n) را از کتاب‌خانه‌ی پیوندی بخواند؛
- به کمک کتاب‌خانه‌ی پیوندی سئوالات مورد نظر را از شرکت هیولاها بپرسد؛
- به کتاب‌خانه‌ی پیوندی گزارش دهد ترتیب هیولاها بر حسب حقوق چگونه است.

کتاب‌خانه

توابع زیر در کتاب‌خانه یافت می‌شود:

`int init()` این تابع n (تعداد هیولاها) را باز می‌گرداند ($1 \leq n \leq 1000$).

`int compare(int i, int j)` این تابع یک پرسش (مقایسه) از هیولاها می‌پرسد و پاسخشان را باز می‌گرداند. دو پارامتر ورودی این تابع، شماره‌ی دو هیولا^۳ است. خروجی این تابع

^۱ زمان مجاز در واقع ۵۰۰ میلی‌ثانیه می‌باشد و کتاب‌خانه حداکثر ۱۰۰ میلی‌ثانیه از آن را استفاده می‌کند.
^۲ حافظه‌ی مصرفی مجاز در واقع ۳۲ مگابایت می‌باشد و کتاب‌خانه حداکثر ۱ مگابایت از آن را استفاده می‌کند.
^۳ شماره‌ها از ۱ تا n هستند.

برابر 1 است اگر حقوق هیولای i کمتر از حقوق هیولای j باشد. در غیر این صورت، خروجی برابر 0 است.

`void report(int *index)` با فراخوانی این تابع می‌توانید پاسخ خود را به کتاب‌خانه تحویل دهید. این تابع باز نمی‌گردد و پس از فراخوانی‌اش برنامه‌ی تان پایان می‌یابد. ورودی این تابع، یک اشاره‌گر است به ابتدای آرایه‌ای به طول n از نوع `int`. شماره‌ی هیولاها را به صورت مرتب شده برحسب حقوق در این آرایه قرار دهید^۴. این آرایه باید شامل جایگشتی از اعداد ۱ تا n باشد.

تابع `init()` باید حتماً قبل از دیگر توابع کتاب‌خانه فراخوانی شود. نقض شرایط گفته‌شده به منزله‌ی نقض استفاده‌ی درست از کتاب‌خانه است. برنامه‌ی تان نباید با هیچ فایل‌ی کار کرده، از ورودی استاندارد بخواند یا به ورودی استاندارد بنویسد. در ضمن برنامه‌ی تان نباید سعی کند که به فضایی خارج از برنامه دسترسی داشته باشد. بدیهی است که نقض هر یک از این حدود می‌تواند منجر به حذف شدن از گردونه‌ی رقابت‌ها گردد.

دو فایل `sort.h` و `sortlib.cpp` به شما داده شده است. بالای برنامه‌ی تان باید عبارت `#include "sort.h"` را قرار دهید. توابع و تعاریفی که توسط کتاب‌خانه در اختیار شما قرار گرفته، به صورت زیر است:

```
int init();
int compare(int i, int j)
void report(int *index)
```

برای هم‌گردانی برنامه‌ی تان فایل‌های فوق‌الذکر را پیش برنامه‌ی تان (در این جا با نام `sort.cpp`) کپی کرده و از دستور

```
g++ -O2 -static sort.cpp sortlib.cpp -lm
```

استفاده کنید. فایل `ssort.cpp` نیز یک نمونه از کار کردن با این کتاب‌خانه است. توجه به دو نکته ضروری است:

۱. برنامه‌ی نمونه‌ی داده شده، یعنی `ssort.cpp` کار مفیدی انجام نمی‌دهد و تنها برای نمایش نحوه‌ی کار کردن با کتاب‌خانه فراهم شده است.

۲. در هنگام ارزیابی برنامه‌ها از کتاب‌خانه‌ی دیگری استفاده خواهد شد که توابع و تعاریف درون `sort.h` را حمایت می‌کند.

آزمایش

کتاب‌خانه‌ی آزمایش را می‌توانید از سایت مسابقات دریافت کنید. کتاب‌خانه‌ی آزمایشی داده شده، داده‌های مورد نیاز را از ورودی استاندارد می‌خواند. باید در سطر اول ورودی، n (تعداد

^۴ شماره‌ی هیولا با کمترین حقوق در `index[0]` و هیولا با بیشترین حقوق در `index[n-1]` قرار می‌گیرد.

هیولاها)، و در سطر بعد، n عدد متفاوت نوشته شده باشند که با فاصله از هم جدا شده‌اند. عدد i ام سطر دوم مقدار حقوق هیولای شماره‌ی i را نشان می‌دهد. هنگام استفاده از واسط آزمایش (TEST) برنامه‌ی تان در برابر همین کتاب‌خانه کار می‌کند و فایلی که به‌عنوان ورودی ارسال می‌کنید، باید دارای ساختار مذکور باشد. شما آزادید برنامه‌هایی که دراختیارتان قرار داده شده را تغییر دهید ولی توصیه می‌شود که فایل `sort.h` را تغییر ندهید.

شیوه ارزیابی

اگر برنامه‌ی تان از محدودیت‌های اعمال شده تجاوز کند، یا از کتاب‌خانه به‌صورت شایسته استفاده ننماید، به‌ازای آن موردِ آزمون، نمره‌ای به برنامه تعلق نمی‌گیرد. برنامه‌تان حداکثر اجازه‌ی انجام ۲۰۰۰۰۰۰۰ پرسش را دارد. اگر بیش از این تعداد پرسش انجام دهید، برنامه‌تان متوقف شده و نمره‌ای نمی‌گیرید. اگر در پایان کار، اتصالات گزارش شده بین پایگاه‌ها نادرست باشد نیز نمره‌ای نمی‌گیرید. برای هر مورد آزمون، یک عدد به عنوان تعداد پرسش‌های مورد نیاز برای یافتن حتمی پاسخ در نظر گرفته‌ایم. اگر این عدد $best$ و تعداد پرسش‌های شما $questions$ باشد، در صورت دادن پاسخ نهایی درست، به اندازه‌ی

$$\min \left\{ 100, 100 \times \frac{best + 1}{questions + 1} \right\}$$

از نمره‌ی آن مورد تست را دریافت می‌کنید.

تعامل نمونه

توضیحات	فراخوانی
مقدار n برابر خروجی تابع یعنی 3 می‌شود.	<code>int n=init();</code>
مقدار x برابر خروجی تابع یعنی 1 می‌شود.	<code>int x=compare(1, 2);</code>
مقدار y برابر خروجی تابع یعنی 1 می‌شود.	<code>int y=compare(1, 3);</code>
مقدار z برابر خروجی تابع یعنی 0 می‌شود.	<code>int z=compare(2, 3);</code>
تعریف یک آرایه از نوع <code>int</code> جهت دادن جواب نهایی.	<code>int a[3];</code>
مقدار دهی a .	<code>a[0]=1; a[1]=3; a[2]=2;</code>
جواب نهایی را اعلام می‌کنیم.	<code>report(a);</code>