

باسمه تعالی

آزمون عملی نخست

ساختمان داده‌ها و الگوریتم‌ها

۲۲ فروردین ۱۳۹۸



لطفا پیش از شروع کار موارد زیر را به دقت مطالعه کنید:

- این آزمون شامل ۳ سوال است که هرکدام ۱۰۰ امتیاز دارند، نمره کامل آزمون ۲۵۰ است و در صورتی که بیش از این میزان نمره کسب کنید طبق ضوابط گفته شده در درس بخشی از آن به صورت نمره امتیازی در نظر گرفته خواهد شد.
- زمان امتحان ۴ ساعت است و در صورت عدم بروز مشکل جدی تمدید نخواهد شد.
- از خواندن همه‌ی سوالات و فکر کردن به میزان کافی بر روی آن‌ها اطمینان حاصل کنید؛ هم چنین بعضی سوالات دارای زیرمسئله‌هایی هستند که بعضاً تا ۶۰ درصد نمره سوال را شامل می‌شوند. پس از کنار زیرمسئله‌ها به سادگی رد نشوید!
- از بین دو آزمون عملی، آن آزمونی که نمره‌ی بیشتری کسب کنید ۳ نمره و آزمون دیگر ۱ نمره از نمره‌ی نهایی شما را شامل خواهند شد.
- دسترسی به اینترنت در طول امتحان قطع است و تنها سایت‌های کوئرا و docs.python.org قابل دسترسی هستند.
- در زبان پایتون اندازه پیش فرض stack برای فراخوانی بازگشتی کم است؛ برای افزایش این مقدار، دو خط زیر را بالای برنامه‌ی خود قرار دهید:

```
import sys
sys.setrecursionlimit(1000000)
```

- ممکن است تست‌ها در طول مسابقه تغییر کنند که در این صورت کد های شما مجددا داوری میشوند و به شما اطلاع داده میشود. هم چنین ممکن است تست‌ها بعد از پایان مسابقه هم تغییر کنند و کد های شما مجددا داوری شوند.
- دقت کنید که تنها ارسال نهایی شما برای هر سوال ملاک نمره دهی میباشد و بعد از پایان امتحان نمی‌توانید ارسال نهایی خود را تغییر دهید.
- مصادیق تقلب در این امتحان عبارت‌اند از:
 - هرگونه مشورت یا گرفتن راهنمایی یا راه‌حل از سایر افراد یا منابع اینترنتی.
 - استفاده از هرگونه وسیله الکترونیکی دیگر مثل لپ‌تاپ، موبایل، ساعت هوشمند، حافظه فلش، سی‌دی یا دی‌وی‌دی یا وسایلی که از اینترنت اشیا استفاده می‌کنند.
 - فرستادن سایر افراد یا عامل‌های هوشمند (agent) یا ربات‌های انسان‌نما به جای خود.
 - آپلود یا دانلود مغز سایر دانشجویان یا مسئولین برگزاری بر روی مغز خود یا بقیه.
 - بازی کردن boardgame به صورت تکی یا دسته‌جمعی
- در صورت ارتکاب تخلف طبق مصادیق بالا نمره‌ی متخلفین از این آزمون صفر خواهد شد و برخوردهای شدیدتری نیز طبق تصمیم تیم تدریس صورت خواهد گرفت.
- استفاده از برخی امکانات زبان پایتون در این آزمون ممنوع است؛ از جمله:
 - تابع‌های الگوریتمی مثل sort, partition و ...
 - داده‌ساختارهای آماده مثل مجموعه، دیکشنری، stack، queue و ... (تنها داده‌ساختار مجاز، آرایه است)
 - توابع یا عملگرهایی که عناصر را در یک گردایه جست‌وجو می‌کنند؛ مثل تابع contains یا index یا عملگر in
 - عملگر == بین رشته‌ها یا آرایه‌ها یا توابعی مثل startswith و غیره.
 - برنامه‌نویسی multi-thread
 - کتاب‌خانه‌ی یادگیری ماشین (sklearn) و تلاش برای پیاده‌سازی عامل‌های هوشمند با قابلیت یادگیری
 - استفاده از توان پردازشی کارت گرافیک (GPU) برای کاهش زمان اجرای برنامه
 - برنامه‌نویسی غیرمتمرکز (decentralized) و تلاش برای پیاده‌سازی بلاک‌چین
 - استفاده از سرویس‌های ابری (cloud) داخلی و خارجی (آمازون، گوگل و ...) و یا پیاده‌سازی یک سامانه ابری
 - پیاده‌سازی بدافزار و آلوده کردن کامپیوترهای سایت به انواع ویروس‌ها
 - امکانات شبکه‌ای پایتون و تلاش برای تغییر آی‌پی، خوانش و یا دستکاری داده‌های رد و بدل شده در شبکه
 - هک کردن سامانه‌ی کوئرا و یا سایت دانشگاه صنعتی شریف
 - پیاده‌سازی سیستم عامل جدید و برنامه نویسی بر روی آن
 - رایانش کوانتومی

افتخارآفرینی سیدپارسا

محدودیت زمان: ۱ ثانیه

محدودیت حافظه: ۲۵۶ مگابایت

سیدپارسا که اصولاً عادت به افتخارآفرینی دارد، این بار افتخار جدیدی آفریده است و به همراه دوستانش علی و مهدی موفق شد پس از ۱۸ سال یک تیم ایرانی را در مسابقات ACM جهانی صاحب مدال کند! سیدپارسا که از این موفقیت بسیار سرمست و شادمان بود فراموش کرد که باید برای همسرش سوغاتی بخرد و مجبور شد به صورت بسیار عجله‌ای از یک دست‌فروش یک درخت بخرد تا برای همسرش ببرد؛ البته متأسفانه اجناس این دست‌فروش بسیار بی‌کیفیت بودند و این درخت خریداری شده در حین حمل و نقل در هواپیما تکه‌تکه شد!

سیدپارسا پس از بازگشت به ایران ناامیدانه به دنبال راهی بود تا بلکه بتواند درختش را تعمیر کند. الیاس که از سالیانی دراز در کار خرید و فروش انواع و اقسام علف و گل و درخت بوده است به کمک سیدپارسا شتافت و به وی گفت اگر در کاتالوگ درخت نوشته شده باشد که هر رأس چند فرزند دارد، با دانستن این اطلاعات می‌توان درخت را بازیابی کرد. سیدپارسا نگاهی به کاتالوگ انداخت و متوجه شد که آنجا تعداد فرزندان تمام رؤوس به ترتیب آمده است؛ ترتیب وارد شدن رؤوس نیز این‌گونه است که ابتدا ریشه، سپس همه‌ی رؤوس ردیف اول (از چپ به راست)، سپس همه‌ی رؤوس ردیف دوم و آمده‌اند. حال در اولین گام الیاس می‌خواهد ارتفاع درخت را تشخیص دهد. به الیاس و سیدپارسا کمک کنید تا آبروی این سید عزیز نزد همسرش حفظ شود!

ورودی

در خط اول ورودی عدد n می‌آید که بیانگر تعداد رأس‌های درخت است ($1 \leq n \leq 10^6$)؛ سپس در خط بعد n عدد می‌آیند که عدد i ام بیانگر تعداد فرزندان رأس i ام (طبق ترتیب گفته‌شده) است. تضمین می‌شود که درختی با این مشخصات وجود دارد.

خروجی

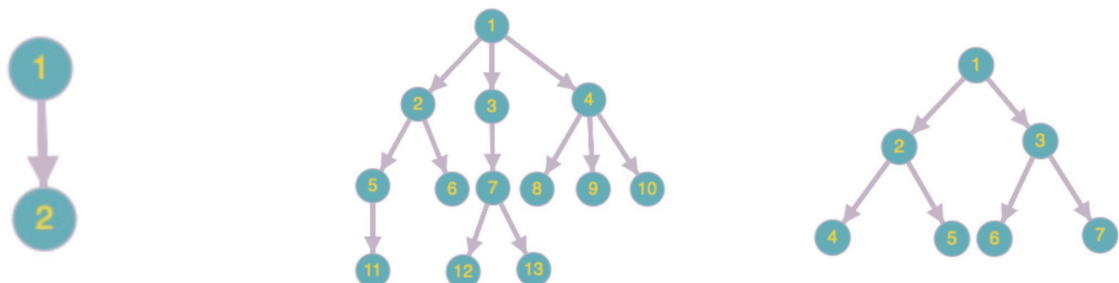
در خروجی ارتفاع درخت را چاپ کنید.

ورودی و خروجی نمونه

ورودی	خروجی
7 2 2 2 0 0 0 0	3
13 3 2 1 3 1 0 2 0 0 0 0 0 0	4
2 1 0	2

توضیحات

در شکل‌های زیر به ترتیب از راست به چپ درخت‌های متناظر با ورودی‌های نمونه اول تا سوم را مشاهده می‌کنید. اعداد روی رأس‌ها بیانگر ترتیب آمدن آن‌ها در ورودی هستند.



دکتر شریفی vs طالبی

محدودیت زمان: ۰/۵ ثانیه
محدودیت حافظه: ۲۵۶ مگابایت

با فرارسیدن لحظه تحویل سال دکتر شریفی طبق یک سنت دیرینه می‌خواست به تمامی تی‌ای‌هایش در درس DS یک پیام تبریک جداگانه بفرستد؛ اما با توجه به این‌که تعداد تی‌ای‌های او در این درس بسیار زیاد بودند و سربه‌فلک کشیده بودند؛ وی تصمیم گرفت تعداد زیادی پیام تبریک آماده از اینترنت پیدا کند و هر پیام را برای یکی از تی‌ای‌هایش بفرستد. البته این کار نباید به صورتی انجام می‌شد که تابلو شود دکتر پیام‌ها را خودش نوشته است!

الیاس که از سالیانی دراز در کار پیام تبریک عید است به دکتر شریفی گفت که هیچ‌یک از پیام‌های تبریک پیشوند (prefix) پیام تبریک دیگری باشد. بنابراین دکتر شریفی از طالبی (تصویر ۱) که برای برگزاری مسابقه DataDays تقریباً به صورت شبانه‌روزی در دانشگاه بود درخواست کرد که یک سیستم (تصویر ۲) پیاده‌سازی کند که در آن پیام‌های تبریک مختلف را یکی‌یکی وارد کند و به محض این‌که به حالتی رسید که یک رشته زیررشته‌ی رشته‌ی دیگری بود و اصطلاحاً قضیه «تابلو» شد به او هشدار دهد. طالبی از آن‌جا که باید کار فنی مسابقه‌اش را انجام دهد نمی‌تواند زمان زیادی برای این درخواست دکتر شریفی اختصاص دهد و بنابراین از شما خواسته تا به جای او این سیستم را پیاده‌سازی کنید!



تصویر ۲: سیستم پیاده‌سازی شده توسط طالبی



تصویر ۱: طالبی

ورودی

در هر خط ورودی یک پیام می‌آید (پیام‌ها به ترتیب اضافه‌شدنشان می‌آیند). هر پیام تبریک رشته‌ای ناتهی شامل حروف کوچک انگلیسی است. تعداد کل خط‌ها مشخص نیست ولی طول پیام‌های تبریک مجموعاً بیشتر از 10^5 نمی‌شود. آخرین خط ورودی تنها شامل یک عدد صفر است که بیان‌گر اتمام ورودی است (این خط جزو پیام‌ها نیست).

خروجی

در خروجی اولین رشته‌ای که اضافه‌شدن آن به مجموعه رشته‌های قبلی باعث می‌شود قضیه «تابلو» شود را چاپ کنید. اگر بعد از اضافه شدن همه‌ی پیام‌ها هم قضیه تابلو نشد، عبارت everything is alright را چاپ کنید.

زیرمسئله‌ها

- در ۳۰ درصد تست‌ها تعداد کل پیام‌ها حداکثر ۱۰۰ است و هر پیام نیز حداکثر ۱۰۰ حرف دارد.
- در ۵۰ درصد تست‌ها (شامل ۳۰ درصد قبلی) طول پیام‌ها به ترتیب اکیداً نزولی است؛ یعنی هر پیام از پیام‌های بعدی خود درازتر مساوی است.

ورودی و خروجی نمونه

ورودی	خروجی
salenomobarak happynewyear heyheyhey saleno h salamsalamsalam doctorsharifi kianoosh 0	saleno
sathsathsathsath seyyedebazorg tabreeek tabrik 0	everything is alright
asghar akb akbar 0	akbar
whatthehellareyoudoing tabestoonkootah mabaeslamchekardim sath dont sath 0	sath

توضیحات

در نمونه‌ی اول پیام saleno پیشوند پیام salenomobarak است. دقت کنید که پیام h هم پیشوند پیام دوم و سوم است ولی شما باید اولین پیامی که اضافه شدنش باعث ایجاد مشکل می شود را چاپ کنید. در نمونه‌ی سوم هم بعد از اضافه شدن پیام akbar، پیام akb که قبلا اضافه شده بود پیشوند این پیام می شود و قضیه تابلو می شود! در نمونه‌ی چهارم دومین پیام sath پیشوند اولین پیام sath (که قبلا اضافه شده) است.

زیدان هم برگشت؛ اما تو نیامدی!

محدودیت زمان: ۱ ثانیه

محدودیت حافظه: ۲۵۶ مگابایت

بنیامین که از طرفداران پروپاقرص تیم رئال مادرید است از شنیدن خبر بازگشت زیدان به رئال مادرید به حدی شادمان شد که تصمیم گرفت روز بعد به تمام افراد لابی دانشکده کامپیوتر شیرینی بدهد! او همان شب به شیرینی فروشی رفت و تعداد بسیار زیادی شیرینی خرید تا روز بعد آن‌ها را پخش کند. برای سادگی می‌توانید فرض کنید که در لابی دانشکده کامپیوتر n نفر حضور دارند که در یک خط از چپ به راست ایستاده‌اند؛ به سمت چپ ترین فرد شماره ۱ و به سمت راست ترین فرد شماره n نسبت می‌دهیم.

بنیامین قصد داشت کار پخش را به m مرحله تقسیم کند و در مرحله i ام یک بازه $[l_i, r_i]$ انتخاب کند و به افرادی که شماره‌شان در این بازه است یک شیرینی بدهد. الیاس که از سالیانی دراز در کار شیرینی فروشی است به بنیامین گفت برای این که از هرگونه درگیری جلوگیری کند باید بازه‌هایش را طوری انتخاب کند یا هیچ اشتراکی با هم نداشته باشند، یا یکی زیربازه‌ی دیگری باشد؛ به بیان دیگر این عبارت ریاضی باید برقرار باشد (نماد \vee به معنی «یا»ی منطقی است):

$$\forall i, j \in [1, m] : [l_i, r_i] \cap [l_j, r_j] = \emptyset \vee [l_i, r_i] \subseteq [l_j, r_j] \vee [l_j, r_j] \subseteq [l_i, r_i]$$

پس از اتمام شیرینی‌ها کیانوش نزد بنیامین آمد و به عنوان جایزه به او یک آرایه به اسم a داد که عنصر a_i بیان‌گر تعداد شیرینی‌هایی بود که نفر i ام خورده. بنیامین که به شدت علاقه‌مند به مباحث بهینه‌سازی بود می‌خواست ببیند برای تولید این آرایه روش بهینه‌ی پخش شیرینی کدام بوده! دقت کنید روشی بهینه است که در آن m کمینه شود و البته این روش بهینه لزوماً یکتا نیست.

ورودی

در خط اول ورودی عدد n می‌آید که بیان‌گر تعداد کل افراد لابی است ($1 \leq n \leq 5 \times 10^4$)؛ سپس در خط بعد n عدد می‌آیند که بیان‌گر آرایه‌ی a هستند ($0 \leq a_i \leq 5 \times 10^4$).

خروجی

در خط اول خروجی عدد m کمینه را چاپ کنید؛ سپس در m خط بعدی به ترتیب بازه‌ها را چاپ کنید. دقت کنید که انتخاب بازه‌ها ممکن است یکتا نباشد و شما هر حالت درست دلخواهی را می‌توانید چاپ کنید. تضمین می‌شود که این m کمینه از 5×10^3 بیشتر نخواهد شد.

زیرمسئله

• در ۶۰ درصد تست‌ها $1 \leq n \leq 500$ و $0 \leq a_i \leq 500$.

ورودی و خروجی نمونه

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