



دانشکده مهندسی کامپیوتر

ساختمان‌های داده

نیم‌سال دوم ۱۳۹۹-۱۴۰۰

مدرس: مریم اسدی

موعد تحویل: ۱۴۰۰/۰۴/۱۲

پروژه

نکات

- ۱- پروژه در دو فاز طراحی شده است. فاز دوم پیش از پایان ترم به شما اعلام می‌شود.
- ۲- انجام پروژه به صورت فردی است.
- ۳- هر بخش از پروژه به همراه یک مستند در نرم‌افزار Microsoft Word که در بردانده‌ی اطلاعات کامل راجع به آن بخش است باید ارائه شود. رعایت موارد زیر در این مستند الزامی است:
 - در مستند پروژه‌ی خود، نحوه دقیق و کامل اجرای برنامه‌ی خود را شرح دهید.
 - سپس الگوریتم پیشنهادی برای پیاده‌سازی بخش‌های مختلف شرح داده شود.
 - فونت متن نوشته شده از یکی از انواع استاندارد فونت باشد (به عنوان مثال برای متن‌های فارسی از فونت‌های Nazanin، B Nazanin، Zar، B Zar می‌توان استفاده نمود)
 - فاصله و نیم‌فاصله رعایت شود.
 - اندازه استاندارد فونت فارسی در متون ۱۴ و فونت انگلیسی ۱۲ خواهد بود.
 - بخش‌های مختلف با عناوین مختلف از یکدیگر جدا شوند.
- ۴- برای پیاده‌سازی پروژه از یکی از زبان‌های برنامه‌نویسی C++، Java، و یا Python می‌توانید استفاده کنید.
- ۵- پروژه‌ی نهایی در قالب یک پوشه‌ی فشرده با عنوان «Name-Student Number» در کوئرا بارگذاری شود.

مقدمه

رستوران یاس را که به خاطر دارید! می‌خواهیم باهم برگردیم به دورانی که یاس جای سوزن انداختن نداشت. در یک روز خواب‌آور بهاری، «فراز»، دانشجوی خسته‌ی مهندسی کامپیوتر، بعد از کلاس ساختمان‌های داده خودش را به رستوران یاس می‌رساند، اما با صف طولانی غذا مواجه می‌شود.

بعد از چند دقیقه ایستادن در صف، فراز متوجه می‌شود که صف یاس یک صف معمولی نیست! هر دانشجو که وارد رستوران می‌شود، ابتدا به دنبال دوستان خود در صف می‌گردد و به جای اضافه شدن به آخر صف، پشت سر هر دسته از دوستانش که از بقیه جلوتر هستند، می‌ایستد. اگر فردی دوستی در صف نداشت، یا مثل فراز از همه‌جا بی‌خبر بود، به ناچار آخر صف می‌ایستد.

فاز اول

۱- صف یاس را با استفاده از ۲ پشته پیاده‌سازی کنید. پیچیدگی زمانی الگوریتم خود را برای ورودی به اندازه‌ی n محاسبه کنید.

۲- برنامه‌ای بنویسید که ساختمان داده‌ی صف یاس را پیاده‌سازی کند. به طوری که در ورودی شماره دانشجویی دانشجویان حاضر در صف را به تفکیک رشته‌ی تحصیلی بگیرد و یک صف با توابع زیر پیاده‌سازی کند:

ENQUEUE (n) -> دانشجویی با شماره دانشجویی n وارد صف می‌شود

DEQUEUE () -> غذای نفر اول صف تحویل داده می‌شود

STOP () -> پایان دستورات

۳- پیچیدگی زمانی الگوریتم خود در قسمت دوم را محاسبه کنید (اختیاری).

ورودی

در خط اول ورودی هر تست کیس، تعداد رشته‌های تحصیلی ($t < 1001$) دریافت می‌شود. در t خط بعدی، تعداد دانشجوی هر رشته (حداکثر ۱۰۰۰) و به تعداد آن، شماره دانشجویی دانشجویان آن رشته (اعدادی بین ۰ تا ۹۹۹۹۹۹) دریافت می‌شود. در آخرین تست کیس، تعداد رشته‌ها صفر داده می‌شود. در نظر داشته باشید که یک تست کیس ممکن است تعداد دستورات بسیار بالایی داشته باشد؛ پس سعی کنید دستورات را به بهینه‌ترین شکل پیاده‌سازی کنید.

خروجی

برای هر تست کیس، در خط اول شماره‌ی تست کیس به صورتِ "Scenario #k" چاپ شود. سپس برای هر دستور "DEQUEUE"، شماره دانشجویی شخصی که غذا گرفته است چاپ شود. بعد از پایان هر تست کیس، یک خط خالی در خروجی قرار داده شود.

مثال

ورودی (سناریوی ۲)

| |
|---|
| 2 |
| 5 259001 259002 259003 259004 259005 |
| 6 260001 260002 260003 260004 260005 260006 |
| ENQUEUE 259001 |
| ENQUEUE 260001 |
| ENQUEUE 259002 |
| ENQUEUE 259003 |
| ENQUEUE 259004 |
| ENQUEUE 259005 |
| DEQUEUE |
| DEQUEUE |
| ENQUEUE 260002 |
| ENQUEUE 260003 |
| DEQUEUE |
| DEQUEUE |
| DEQUEUE |
| DEQUEUE |
| STOP |
| 0 |

ورودی (سناریوی ۱)

| |
|---------------|
| 2 |
| 3 101 102 103 |
| 3 201 202 203 |
| ENQUEUE 101 |
| ENQUEUE 201 |
| ENQUEUE 102 |
| ENQUEUE 202 |
| ENQUEUE 103 |
| ENQUEUE 203 |
| DEQUEUE |
| DEQUEUE |
| DEQUEUE |
| DEQUEUE |
| DEQUEUE |
| DEQUEUE |
| STOP |
| 0 |

خروجی (سناریوی ۲)

| |
|-------------|
| Scenario #2 |
| 259001 |
| 259002 |
| 259003 |
| 259004 |
| 259005 |
| 260001 |

خروجی (سناریوی ۱)

| |
|-------------|
| Scenario #1 |
| 101 |
| 102 |
| 103 |
| 201 |
| 202 |
| 203 |

فاز دوم

بخش اول:

در فاز اول پروژه تا حدودی با رستوران یاس و صف آن آشنا شدید. بعد از آن اتفاقات، دانشجویان متوجه می‌شوند که یک مسئول حراست به‌عنوان انتظامات رستوران، داخل رستوران حضور دارد تا از چنین بی‌نظمی‌هایی جلوگیری کند. این مسئول شایسته تمام کار خود را از طریق کامپیوترش انجام می‌دهد. برای رسیدگی به تخلفات مربوط به رعایت ترتیب در صف، کامپیوتر او، به‌عنوان ورودی، صف را به‌صورت یک لیست پیوندی دو طرفه دریافت کرده و می‌تواند دو متد زیر را روی ورودی اجرا کند:

(۱) متد `getStudent` که یک اندیس بگیرد و شماره دانشجویی دانشجوی حاضر در آن اندیس را در خروجی چاپ کند.

(۲) متد `swap`، که با دریافت دو اندیس، ترتیب قرار گرفتن دانشجویان بین آن دو اندیس را معکوس کند و به این ترتیب، (به نظر خودش) آن‌ها را تنبیه کند.

سعی کنید برنامه‌ای که آقای انتظامات از آن برای این کار استفاده می‌کند را خودتان بنویسید. برنامه‌ی شما باید حداقل متد های زیر را پیاده‌سازی کند.

شماره دانشجویی دانشجوی حاضر در اندیس i را برگرداند. $\rightarrow \text{getStudent}(\text{int } i)$

ترتیب دانشجویان از a تا b را معکوس کند. $\rightarrow \text{swap}(\text{int } a, \text{int } b)$

ورودی :

در خط اول تعداد دانشجویهای حاضر در صف وارد می‌شود که به‌عبارتی همان اندازه‌ی لیست پیوندی است ($n < 10^6$). سپس در n خط بعدی، شماره دانشجویی دانشجویهای حاضر در صف وارد می‌شود. در خط بعدی تعداد تخلف‌ها وارد می‌شود ($m < 10^4$) و بعد در m خط بعدی، یکی از دو ورودی `GET i` و `SWAP a b` که متدهای برنامه شما را فراخوانی می‌کنند، به‌صورت رشته وارد می‌شود.

خروجی:

به ازای هر دستور `GET i` در ورودی، شماره دانشجویی دانشجوی i ام چاپ می‌شود.

| خروجی نمونه | ورودی نمونه |
|-------------|-------------|
| 333 | 9 |
| 444 | 111 |
| | 222 |
| | 333 |
| | 444 |
| | 555 |
| | 666 |
| | 777 |
| | 888 |
| | 999 |
| | 3 |
| | GET 3 |
| | SWAP 1 6 |
| | GET 3 |

بخش دوم:

فراز که خیلی ول نکن است، در صدد اعتراض به مسئول مربوطه در دانشگاه بر می آید و از انتظامات در این مورد سؤال می کند. اما کامپیوتر انتظامات فقط می تواند از روی پیمایش پس ترتیب درخت دودویی مسئولین دانشگاه، مسئول مورد نظر را پیدا کند و آقای انتظامات، فقط پیمایش های پیش ترتیب و میان ترتیب را در اختیار دارد.

حال شما باید برنامه ای بنویسید که از روی پیمایش پیش ترتیب و میان ترتیب یک درخت دودویی، بتواند پیمایش پس ترتیب آن را برگرداند.

ورودی:

در خط اول تعداد رئوس درخت وارد می شود ($1 \leq n \leq 10^4$). در هر یک دو خط بعدی، n عدد به تفکیک فاصله (space) وارد می شوند که به ترتیب نشان دهنده ی پیمایش های پیش ترتیب و میان ترتیب درخت هستند.

خروجی:

n عدد به تفکیک فاصله در یک خط چاپ می شوند که نمایش گر پیمایش پس ترتیب درخت دودویی مورد نظر هستند.

| نمونه ورودی | نمونه خروجی |
|---|-------------------|
| 9 6 2 1 4 3 5 7 9 8 1 2 3 4 5 6 7 8 9 | 1 3 5 4 2 8 9 7 6 |