باسمه تعالی

داده‌ساختارها و مبانی الگوریتم‌ها (۴۰-۲۵۴)

Data Structures and Fundamentals of Algorithms

دانشکده‌ی مهندسی کامپیوتر

مدرس: محمد قدسی

تمرین شماره‌ی ۲ — موعد ارسال: ۱۳۹۲/۸/۳

۱. با کمک یک آرایه به اندازه‌ی N دو پشته پیاده‌سازی کنید که به طور همزمان، هر دو از عملیات push و pop پشتیبانی کنند. الگوریتمی که ارائه می‌دهید فقط وقتی که مجموع اندازه عناصر دو پشته بیشتر از N باشد، باید خطای Stack Over Flow را اعلام کند. شبه کد الگوریتم خود را بنویسید.

۲. همانطور که می‌دانید الگوریتم Insertion Sort به صورت زیر است:

Algorithm 1 Insertion Sort

1: **for** $j \leftarrow 2$ to $length[A]$ **do**
2: Find position of $A[j]$ in sorted subArray $A[1]$ to $A[j-1]$
3: Insert $A[j]$ into Founded position
4: **end for**

در کلاس به شما گفته شده این الگوریتم از $\mathcal{O}(n^2)$است. اگر بتوانیم خطهای ۲ و ۳ را طوری تغییر دهیم که در زمان بهتری انجام شوند، زمان کل الگوریتم بهتر خواهد شد.

الف) با کمک جستجوی دودویی، زمان اجرای خط ۲ را بهتر کنید. آیا اگر اعداد را در آرایه‌ای نگه داشته باشیم، زمان کلی الگوریتم بهتر خواهد شد؟

ب) فرض کنید اعداد در داده‌ساختار لیست پیوندی نگه داشته شده‌اند. زمان کل الگوریتم را تحلیل کنید.

۳. یک لیست پیوندی از اعداد داریم که در آن هر عضو تنها به عضو بعدی دسترسی دارد. روشی کارا ارائه دهید که بدون تغییر در ساختار لیست و تنها با استفاده از $\mathcal{O}(1)$ حافظه اضافی مشخص کند آیا در این لیست دور وجود دارد یا خیر (توجه کنید که لازم است ساختار لیست در طول اجرای الگوریتم ثابت بماند و یکسان ماندن آن تنها در ابتدا و انتها کافی نیست).

۴. یک لیست پیوندی یکطرفه با N گره داریم. یکطرفه بودن یعنی برای هر گره آن، فقط اشاره‌گر به گره بعدی آن را داریم. الگوریتمی ارائه دهید که با پیش‌پردازشی در زمان $\mathcal{O}(N)$ تغییری در لیست ایجاد کند، که بتوان در $\mathcal{O}(\sqrt{N})$ به عنصر قبلی هر عنصر دسترسی داشت. شما فقط مجاز به استفاده از $\mathcal{O}(\sqrt{N})$ حافظه‌ی جدید هستید.

۵. یک شمارندهٔ k بیتی را در نظر بگیرید (بیت با اندیس ۰ کم ارزش‌ترین بیت است) که مقدار اولیهٔ آن ۰ است. تنها عملی که بر روی این داده ساختار انجام می‌شود، افزایش ۱ واحد به آن است. شبه کد رویه increment (عمل افزایش) را بنویسید.

الف) ثابت کنید هزینهٔ هر عمل افزایش به طور سرشکنی $\mathcal{O}(1)$ است.

ب) نشان دهید اگر عمل decrement را به شمارنده اضافه کنیم دیگر هزینهٔ سرشکنی لزوماً $\mathcal{O}(۱)$ نیست. ولی اگر از سیستم نمایش دیگری که در آن هر رقم ۰، ۱ یا ۲ است استفاده کنیم، می‌توان این دو عمل را طوری پیاده سازی کرد که هزینهٔ سرشکنی هر یک ثابت باشد.

۶. داده ساختار آرایه پویا[۱] را در یکی از زبان‌های C++ یا Java پیاده‌سازی کنید[۲]. مشخص است که استفاده از داده‌ساختارهای این زبان‌ها مجاز نیست و شما فقط با کمک آرایه‌ها و متغیرهای ساده باید این برنامه را پیاده‌سازی کنید.

الف) داده ساختار آرایه پویا را طوری پیاده‌سازی کنید که از دستورات زیر بصورت سرشکن در $\mathcal{O}(۱)$ پشتیبانی کند.

ورودی و خروجی: در خط اول ورودی عدد t تعداد دستورهایی که به برنامه‌ی شما داده خواهد شد، می‌آید. در t خط بعدی، در هر خط، یکی از دستورات زیر خواهد آمد.

- Push back a: عدد a را به انتهای آرایه اضافه می‌کند و مقدار Capacity و Size آرایه را بروز می‌کند.
- Size: اندازه‌ی آرایه را در خروجی چاپ کنید.
- Capacity: ظرفیت آرایه را در خروجی چاپ کنید.
- Pop back: عنصر انتهایی آرایه را پاک می‌کند و مقدار Capacity و Size آرایه را بروز می‌کند.
- Get i: عنصر iام آرایه را در خروجی چاپ می‌کند ($۰ \leq i < size$).

دقت کنید که Capacity در ابتدا ۱ است و اگر هنگام push نسبت $\frac{Size}{Capacity} = ۱$ باشد، Capacity دو برابر خواهد شد. همچنین هرگاه بعد از پاک کردن $\frac{Size}{Capacity} < ۱/۴$ باشد، Capacity نصف خواهد شد (دقت کنید که Capacity اندازه آرایه‌ای است که عناصر در آن نگهداری می‌شوند).

ورودی و خروجی نمونه:

---

[۱] Dynamic Array

[۲] برای آشنایی بیشتر با این داده ساختار می‌توانید به این لینک مراجعه کنید: http://en.wikipedia.org/wiki/Dynamic_array

stdin	stdout
10 Capacity Size Push back 1 Size Push back 2 Push back 3 Get 1 Capacity Size	1 0 1 2 4 3
10 Push back 1 Push back 2 Push back 3 Push back 4 Push back 5 Get 4 Capacity Pop back Get 3 Capacity	5 8 4 8

محدودیت‌ها:

- $۱ \leq t \leq ۱۰^۶$
- تمام اعداد ورودی در بازه $[-۱۰^۹, ۱۰^۹]$ هستند.

ب) در این قسمت، داده ساختار قسمت (الف) را به‌گونه‌ای تغییر دهید که از دستورات زیر نیز بصورت سرشکن در $\mathcal{O}(۱)$ پشتیبانی کند.

- Push front a: عدد a را به ابتدای آرایه اضافه می‌کند و مقدار Capacity و size آرایه را بروز می‌کند.
- Pop front: عدد ابتدایی آرایه را پاک می‌کند و مقدار Capacity و size آرایه را بروز میکند.

ورودی و خروجی نمونه:

stdin	stdout
9 Push back 1 Push front 2 Push back 3 Pop front Get 0 Get 1 Push front -1 Get 1 Capacity	1 3 1 4
11 Push back 1 Push back 2 Pop front Push front 3 Pop back Push back 4 Get 0 Get 1 Pop front Get 0 Get 1	3 4 4 4

محدودیت‌ها:

- $۱ \leq t \leq ۱۰^۶$
- تمام اعداد ورودی در بازه $[-۱۰^۹, ۱۰^۹]$ هستند.

۷. الف) شهر اتوپیا، N کوه با ارتفاع‌های $h_۱, h_۲, \ldots, h_N$ دارد. می‌خواهیم در این شهر یک خط تله‌کابین راه‌اندازی کنیم به طوری که از روی k کوه متوالی عبور کند. برای این کار، باید با خاک‌برداری ارتفاع **همه‌ی** این کوه‌ها را برابر کنیم. اگر ارتفاع ثانویه‌ی کوه‌ها را $d_۱, d_۲, \ldots, d_n$ بنامیم؛ هزینه‌ی ساخت تله‌کابین از کوه tام تا کوه $t+k-۱$ام، برابر است با:

$$\sum_{i=t}^{t+k-۱} h_i - d_i$$

الگوریتمی ارائه دهید که در $\mathcal{O}(n)$ کمترین هزینه‌ی ساخت تله‌کابین را مشخص کند.

ب) الگوریتم خود را برای قسمت (الف) پیاده‌سازی کنید.

ورودی

در خط اول ورودی دو عدد N و k آمده‌اند. در سطر بعدی N عدد طبیعی آمده است که عدد iام معرف ارتفاع کوه iام می‌باشد.

خروجی

در تنها خط خروجی کمترین میزان هزینه برای ساخت تله‌کابین را بنویسید.

ورودی و خروجی نمونه:

stdin	stdout
4 7 14 8 17 2 7 23 1	26
3 7 14 8 17 2 7 23 1	15

محدودیت‌ها:

- $۱ \leq k \leq n \leq ۱۰^۶$
- تمام اعداد ورودی در بازه $[-۱۰^۹, ۱۰^۹]$ هستند.