

میان زم DS استاد صدیقین شماره دانشجویی: ۹۹۱۰۲۰۷۰ تحلیل و طراحی

سؤال (۱) می‌خواهیم یک صف را با استفاده از دو اسک پیاده‌سازی کنیم و نشان دهیم هزینه سرنگ هر عمل  $enq$  یا  $deq$  برابر  $O(1)$  است. (با استفاده از تابع پیاپی)

دو اسک  $A$  و  $B$  را فرض می‌کنیم با این شرایط که برای عمل  $enq(n)$ ، فقط در اسک  $A$ ، عمل  $push(n)$  را انجام می‌دهیم و برای عمل  $deq(n)$ ، به این صورت عمل می‌کنیم که اگر  $B$  خالی نبود، فقط عمل  $pop(n)$  را در اسک  $B$  انجام می‌دهیم و اگر  $B$  خالی بود، همه اعضای اسک  $A$  را به ترتیب  $pop$  می‌کنیم و آن‌ها را در اسک  $B$   $push$  می‌کنیم. حالا اگر  $B$  خالی نبود، اولین عنصر آن را  $pop$  می‌کنیم و به صورت خروجی تحلیل می‌دهیم.

نحوه پیاده‌سازی (۱) که بالا توضیح دادم به این صورت است که از اسک دوم استفاده می‌کنیم تا این موضوع که (عنصری که اول به صف وارد می‌شود وضعیت متفاوتی نسبت به عنصری که اول به اسک وارد می‌شود دارد) را اصلاح کنیم. پس با  $pop$  و  $push$  عناصر در اسک دوم و خروجی گرفتن از اسک دوم، اصلاح کردیم مشکل را.

حالا تحلیل با تابع پیاپی را داریم:

برای این کار، تابع پیاپی را دو برابر اندازه اسک  $A$  می‌گیریم.

هزینه هر مرحله

برای  $enq$  خواهیم داشت:

$$\hat{C}_i = C_i + \Phi_i - \Phi_{i-1} = 1 + 2 \text{len}(A) - 2(\text{len}(A) - 1) = 3$$

پس هزینه سرنگ ۳ است و از  $O(1)$  است.

برای  $deq$  نیز دو حالت داریم  $\Leftarrow$  ۱. اگر اسک پر باشد:

$$\hat{C}_i = 1 + 2 \text{len}(A) - 2 \text{len}(A) = 1 = O(1)$$

$$\hat{C}_i = C_i + \Phi_i - \Phi_{i-1} = 1 = O(1)$$

$\xrightarrow{2 \text{len}(A)}$   
 $\xleftarrow{2 \text{len}(A)}$

۲. اگر اسک خالی باشد:

در نتیجه هزینه سرنگ آن ۱ است و از  $O(1)$  می‌باشد.

شماره دانشجویی : ۹۹۱۰۲۰۷

ایمان محمدی

سؤال ۱) بیانیس پیوستنی و پیوستنی یک درخت به ترتیب به صورت  $ABC/-AK/L-*$  و  $*-A/BC-/AKL$  می باشد. درخت عبارت مربوط به این هارا رسم می کنیم و متن میانی پیوستنی پرانتز گذاری شده آن را بنویسید.

۱. اول متن میانی پیوستنی پرانتز گذاری شده آن را بنویسید می کنیم  $\Rightarrow$

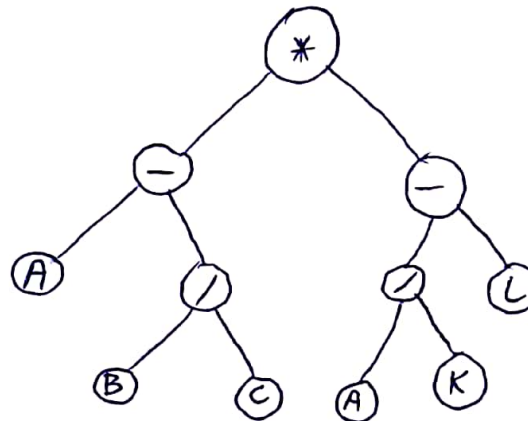
$ABC/-AK/L-*$  و  $-A/BC-/AKL$

inorder

$((A-(B/C)) * ((A/K)-L))$

با استفاده از بیانیس

۲. برای ساخت درخت ، از بیرونی ترین پرانتز شروع می کنیم و عملگر را در رسته قرار می دهیم و به طور بازگشتی از سمت چپ عملگر رسته زیر درخت چپ و از سمت راست عملگر رسته زیر درخت راست را می سازیم. تا ... به عملوند که برگ های درخت هستند می رسم در آخر



سؤال ۳) بخش الف - شرح سؤال شماره پروژه دوم است. اسک  $B$  ای تعریف می کنیم که داده های قادر آن به صورت  $A[i]$  و  $i$  به صورت دو تایی ذخیره می شوند. آرایه  $C$  ای تعریف می کنیم که فرقی فایزای هر اندیس است. در ابتدای برنامه، در اسک  $B$ ، داده تایی  $(5, 10)$  را یوس می کنیم و

همیشه  $1 \leq i \leq n$  برای  $i$  این صورت می رویم که هر بار عنصر بالایی اسک  $B$  را به  $A[i]$  مقایسه می کنیم، اگر  $A[i]$  بزرگتر بود از عنصر آخر اسک  $B$ ، داده دو تایی  $(A[i], B)$  را در  $B$ ، یوس می کنیم و در غیر این صورت، یک بار از  $B$ ، یاب می کنیم و  $A[i]$  را با عنصر بعدی مقایسه می کنیم. این کار را تا آنقدر ادامه می یابیم تا اسک از  $A[i]$  کوچک تر باشد یا اسک  $B$  خالی شود و همه عنصرهای آن یاب شده باشند. اگر این اتفاقات بیفتد، داده تایی  $(A[i], B)$  را در آن یوس می کنیم، برای محاسبه خرجی، می دانیم که در هر مرحله  $A[i]$  برابر است با اندیس ذخیره شده در عنصر بالایی اسک  $B$  قبل از یوس کردن  $(A[i], B)$  در آن و همیشه اگر حالتی که  $B$  خالی می شود، پس باید  $1 \leq i \leq n$

خواهد بود. مقایسه با بالایی اسک و  $push$  به یاد آن  
محاسبه هزینه سرسنگ: برای هر  $A[i]$  در هنگام یوس کردن در مرحله اول، ۲ واحد پرداخت می کنیم و ۱ واحد ذخیره  
مقایسه با بالایی اسک و  $pop$  عنصر بالایی اسک  $B$   
در حالت دوم، در ازای هر  $pop$  از اسک  $B$ ، ۱ واحد ذخیره شده را پرداخت می کنیم

در نتیجه برای انجام هزینه های سرسنگ، همواره موفق خواهیم بود  $\Rightarrow$   
 $O(1) = 2 = \text{هزینه سرسنگ}$



امکان محتمل

۹۹۱۰۲۲۷

ساره دانشجوئی :

سوال ۱۳ - بخش اول - برای حل این بخش از سوال ابتدا یک پار الگوریتم الف را از سمت راست و

یک بار از سمت چپ روی ابرم آرایه اجرا می کنیم. آنقدر انجام می دهیم تا اولین عددی که سمت چپ

و راست یک عدد از آن کوچکتر نباشد. بعد از انجام برای هر عدد، بزرگترین بازه ای که اعداد آن،

بزرگتر یا مساوی عدد هستند را داریم. حالا یک حلقه روی آرایه می زنیم و در هر مرحله، عدد را در طول بازه ای

که همین بالا گفتیم، ضرب می کنیم. حالای آرایه و تغییر  $max$  را به گونه ای تعریف می کنیم که در هر مرحله

اگر ضرب مورد نظر، بزرگتر از  $max$  قبلی بود، مقدار بزرگتر را برابر  $max$  قرار می دهیم. در نهایت

$max$  را، تعداد بزرگترین که می توانیم بخوریم را نشان می دهد.

برای مرتبه نثر داریم که الگوریتم بخش الف، دو بار تکرار می شود و حلقه نثر یک بار  $\Rightarrow$

$$O(n) = مرتبه \Rightarrow 2 O(n) + O(n) = 3 O(n) \quad \text{؛ مرتبه}$$

نمبره دانشجویی : ۹۹۱۰۲۲۰۷ اعمال محمدی  
ما راهمائی استاد

سؤال ۱۴  $m$  آرای مرتب شده است که تعداد کل اعداد دهم مرتبه در آن ها ،  $n$  است .

پرای حل سوال از یک هرم میبینیم (کمینه) استفاده می کنیم به این صورت که کوچکترین عضو آرایه

min heap

از م تا آرایه را در آں یوسا می کشیم و سپس آن عطفورا از آرایه اولیه سنان حذف می کنیم.

حالا متغیر  $\min$  را کوچکترین مقدار داخل هرم و متغیر  $\max$  را بزرگترین مقدار داخل هرم، تعریف می کنیم.

و ہمیشہ اختلاف اربع دو مقدار را ، متغیر  $range$  تعریف می کنیم۔  
 $range = \max - \min$

حالا در هر مرحله رسته هم میشیم راجا بساجی کشیم و کوچیکترین عنقر آرایه ای که عدد ۲۲۲ شدن از آن

آمدہ بود را در هر موشیم  $push$  می کنیم و او آرایه آن حذف می کنیم و دوباره  $min$  و  $max$

رادر صورت تغییر مقادیر منتهیم و عالسیم هم، تعریف جدید می کنیم و همیشه اگر این دو مقدار قدر مساوی

تغییر کرد، اگر  $\text{range}$  از مقدار حال حاضر  $\text{max-min}$  بیشتر بود، دوباره  $\text{range}$  را برابر  $\text{max-min}$  قرار

می دهیم، انقدر این کار را ادا می دهیم که یکی از ما آرد به خانه سدر که در اینجا، به چوب

عود نقران می رسم و الگو نیم تمام می شود range را داریم!

از نظر زمانی، بدترین حالت زمانی است که تعداد مراحل  $n$  باشد و در هر مرحله یک  $push$  و  $pop$

برای هر  $n$  عددی داریم که از اوردِر  $O(\log n)$  است. پس مرتبه کلی جواب،  $O(n \log n)$  است!

لے ۲ بار انجام دیں۔

سؤال ۵) یک درخت رسته داریم. با الگوریتمی خواهیم یا پدرش پردازش  $O(n)$  به ازای هر دو رأس مستقیم کند  
آیا یکی جد دیگری هست یا نه  
پیمایش Preorder  $\leftarrow$  از مرتبه  $O(n)$

پس برای اینکه هر رأس درخت باید دو مقدار اضافی ذخیره کند! به روش های زیر آن ها را محاسبه می کنیم:

یک متغیر زمان فرضا می کنیم، (زهره) پیمایش preorder روی درخت با هر حرکت از پدر به  
فرزند یا از فرزند به پدر، مقدار این متغیر زمان را یکی افزایش می دهیم.

هر بار هم که از یک رأس به فرزندش یا پدرش می رویم، قبل از آن که برای اولین بار به یکی از فرزندان  
لے یا اینکه پیمایش به کل تمام شود در خصوص رسته

بریم یا قبل از اینکه از رأس به پدرش برویم، آن متغیر زمان را در رأس ذخیره می کنیم. در نتیجه در پایان،  
لے مقدارش را

زمان های ورود و خروج هر رأس ثبت می شده می دانیم در بین دو زمانی که در هر رأس ذخیره کرده ایم،

در حال  $visit$  کردن فرزندان این رأس هستیم پس می دانیم بعد از پایان پیمایش، یا توجه به

مقادیر ذخیره شده، هر رأسی که زمان ورود و خروجش بین زمان ورود و خروج رأس دیگری باشد،

از نوادگان آن رأس می باشد.

محول به ازای هر رأس، آن رأس و پدرش یک بار پیمایش می شوند، از آن رو  $O(n)$  می باشد

پس حکم ثابت شد ✓



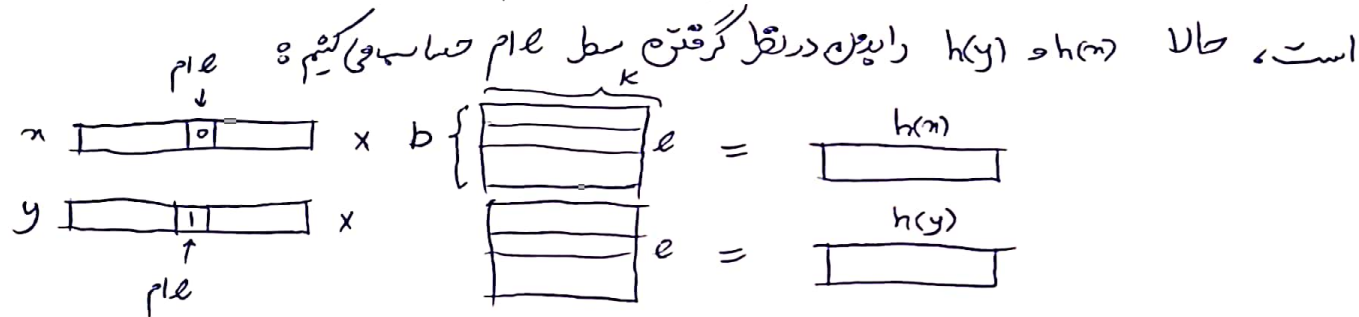
سؤال ۶) در یک درهم سازی، کلیدها مجموعه اعداد  $b$  بیت است و توابع درهم سازی هر کلید را به یک عدد  $k$  بیت تبدیل می کنند.  $m = 2^k$  نشان می دهیم ابرج خانواده از توابع درهم سازی، یک خانواده جهانی است. مجموعه تمام فایده های  $b \times k$  شامل درایه های صفر و یک

$$\forall n, y \in U, h \in H \quad P[h(n) = h(y)] \leq \frac{1}{N} = \frac{1}{2^k} \quad \text{چون } N = 2^k = m$$

پای است، دو کلید  $n$  و  $y$  را متفاوت در نظر می گیریم  $n \neq y$

از آنجایی که ابرج دو کلید متفاوتند، پس حداقل در یک بیت با درایه متفاوتند، فرض می کنیم ابرج بیت به ام باشد، پس خواهیم داشت:

بیرون کم کردن افراده های مسئله، فرض می کنیم بیت  $l$  ام  $n$ ، برابر ۰ و بیت  $l$  ام  $y$  برابر ۱



می دانیم سط  $l$  ام روی  $h(n)$  تأثیری ندارد زیرا همواره در صفر ضرب می شود و برای محاسبه  $h(y)$  مؤثر می باشد.

حالا سط  $l$  ام را محاسبه می کنیم به صورتی که  $h(n) = h(y)$  شود:

$$\begin{bmatrix} \square & \square & \dots & \square \end{bmatrix}_l = \begin{bmatrix} \square & \square & \dots & \square \end{bmatrix}_{h(n)} \text{ XOR } \begin{bmatrix} \square & \square & \dots & \square \end{bmatrix}_{h(y)}$$

در نتیجه داریم که  $l$  یکسان است و یکتا حساب می شود.  $l$  می تواند  $2^k$  حالت مختلف داشته باشد که

$$P[h(n) = h(y)] = \frac{1}{2^k} \quad \leftarrow \text{فقط در یک حالت یکتا } h(n) = h(y) \text{ است}$$