

ب نام آنکه جان را فکرت آموخت



تمرین سری دوم درس ساختمان داده‌ها

استاد درس: دکتر حسینی

موعد تحویل: 1400/09/15



1) صحیح یا غلط بودن گزینه های زیر را با ذکر دلیل مشخص کنید. (پاسخ دادن به 5 مورد از آنها کفایت میکند) ((هر مورد از بارم 10 نمره برخوردار است))

الف) هزینه اضافه کردن یک عنصر جدید به یک لینک لیست برابر $O(N)$ است. درست. بدترین حالت اضافه کردن یک نود به انتهای یک لینکدلیست ساده است که برای این کار باید یک بار کامل بر روی لینکدلیست پیمایش کنیم.

ب) درج یک عنصر در موقعیت فعلی را میتوان به صورت کارا بر روی یک لیست پیوندی یک سویه انجام داد. غلط. برای درج یک عنصر در موقعیت فعلی نیازمند اشاره گر نود قبلی خواهیم بود که برای این کار بایستی یک بار از ابتدا تا یک خانه عقبتر از موقعیت فعلی را پیمایش کنیم که این کار در بدترین شرایط از $O(n)$ می باشد.

ج) هزینه حذف آخرین عنصر از یک لیست یک سویه با دو اشاره گر f و r که به ترتیب به عناصر اول و آخر لیست اشاره میکنند از $O(1)$ است. غلط. برای حذف یک نود در لینکدلیست یک سویه نیازمند اشاره گر $next$ نود قبلی هستیم لذا در یک لینکدلیست یک طرفه ابتدا نیازمند یکبار پیمایش بر روی لینکدلیست خواهیم بود و دو اشاره گر مذکور دردی را دوا نمیکند. (

د) با یک پشته و یک متغیر نمیتوان ساختمان داده ای طراحی کرد که هر بار بتواند اعمال pop $push$ و $return min(stack)$ را پوشش دهد. غلط. توضیحش رو در کلاس حل تمرین داده شد. (

ه) دو آرایه ای دو بعدی از مراتب $m.n$ و $n.p$ داریم. آیا میتوان آرایه ای جدید که حاصل ضرب عناصر آنها (به صورت ماتریسی) است را در اردر کمتر از $O(mnp)$ باشد. (به منظور راحتی کار میتوانید آرایه ها را $n.n$ در نظر بگیرید. صحیح. درواقع میخواهیم دو ماتریس رو در هم دیگه ضرب کنیم. اگر از ضرب استراسن ($strassen\ multiplication$) استفاده کنیم هزینه ای این کار چیزی حدود $(n^{2.8})$ خواهد شد ((در حالت معمولی n^3 است))

ی) در یک لیست حلقوی از $O(1)$ میتوان به ابتدا و انتهای لیست عضو افزود.
 پاسخ این سوال بسته به فرضیات شما دارد. اگر لینکدلیست را یکطرفه در نظر بگیریم پاسخ مسئله **غلط** و اگر دوطرفه در نظر بگیریم پاسخ مسئله **صحیح** مورد قبول است. (توضیحات و فرضیات باید ذکر شده باشند)

2) برای هر یک از چهار نوع لیست در جدول زیر، بدترین حالت مجانبی زمان اجرا برای هر عملیات فهرست شده چقدر است؟
(30 نمره)

	Unsorted, Singly Linked List	Sorted, Singly Linked List	Unsorted, Doubly linked	Sorted, Doubly linked
Search(L,K)	n	n	n	n
Insert(L,K)	1	1	1	1
Delete(L,K)	n	n	1	1
Successor(L,K)	n	1	n	1
Predecessor(L,K)	n	n	n	1
Minimum(L)	n	1	n	1
Maximum(L)	n	n	n	1

3) رویه‌ی روبرو بمنظور وارون کردن یک لیست L طراحی شده است. در مورد درستی روند آن توضیحی مختصر بنویسید. **(10 نمره)**

Reverse (L)

```

if L = null
    then return L
r ← Reverse ( next [L] )
next [ next [ L ] ] ← L
next [ L ] ← null
return r
    
```

پاسخ:

با کمی دقت میتوان دریافت که رویه‌ی بالا همواره در آخرین خانه از لینکدلیست L ((توی خط if L = null)) همواره مقدار null را برمیگرداند و وارد شرط میشود که همان null را ریترن میکند و این مقدار در r ذخیره میشود ((در خط r ← Reverse ()))
 next [L] این مقدار در اولین خانه‌ی لینکدلیست r خواهد نشست؛ در نتیجه طبق تحلیل بالا پاسخ این است که رویه‌ی یاد شده درست کار نمیکند:)

4) به سوالات زیر پاسخ دهید. (20 نمره)

الف) نشان دهید که چطور با دو صف می توان یک پشته ساخت. در مورد پیچیدگی زمانی عملیات های پشته توضیح دهید
عملیات enqueue مانند اضافه کردن یک عنصر به استک اول خواهد بود که این عمل از $O(1)$ است. برای dequeue یک عنصر از استک دوم پاپ میکنیم. اگر استک دوم خالی بود، عناصر استک اول را ابتدا پاپ میکنیم و سپس به استک دوم اضافه میکنیم و در نهایت یک عنصر از استک دوم پاپ میکنیم که این عملیات در بدترین حالت $O(n)$ است.
ب) نشان دهید که چطور با دو پشته می توان یک صف ساخت. در مورد پیچیدگی زمانی عملیات های پشته توضیح دهید.
روش زیر روشی برای پیاده سازی یک پشته با استفاده از دو صف است که در آن پاپ کردن زمان خطی و فشار زمان ثابتی نیاز دارد. بقیه این راهها شامل قرار دادن هر عنصر در صف enqueue کردن آن به هنگام اضافه کردن آن است. سپس، برای انجام پاپ، هر عنصر را از یکی از صف ها dequeue میکنیم و در دیگری قرار می دهید، اما درست قبل از آخرین عنصر متوقف می شوید. سپس، عنصر واحد باقی مانده در صف اصلی را برگردانید.

5) در یک دنباله از اعداد نامرتب، «مقدار نظیر» یک زیردنباله عبارت است از حاصل ضرب طول زیردنباله در کوچک ترین عضو زیردنباله. برای یک دنباله ی دلخواه، زیر دنباله ی متوالی ای را بیابید که مقدار نظیر آن، از همه ی زیر دنباله های متوالی دیگر بیشتر باشد. (از مرتبه زمانی $O(N)$)
مثال: در دنباله ی $\{6, 1, 5, 4, 5, 2, 6\}$ زیر دنباله ی متوالی $\{5, 4, 5\}$ بیشترین مقدار نظیر را دارد. این مقدار برابر است با (20 نمره)

پاسخ:

اگر به ازای هر عضو بتوانیم زیردنباله ی متوالی که در آن این عضو کوچکترین عضو آن زیر دنباله است را به دست بیاوریم مسئله به راحتی قابل حل است (در این حالت حاصل این زیردنباله برابر مقدار آن عضو در طول زیر دنباله است. حال کافی است به ازای تمام زیر دنباله ها که تعداد آنها n تا است این مقدار را محاسبه کرده و max بگیریم)

طبق راه سوال "برج های نیویورک" که در کلاس حل تمرین بیان شد می توانیم با استفاده از استک و دو مرتبه پیمایش (یکبار از سمت چپ و یکبار از سمت راست) این مسئله را در $O(N)$ حل کنیم بدین صورت که :

We traverse all bars from left to right, maintain a stack of bars. Every bar is pushed to stack once. A bar is popped from stack when a bar of smaller height is seen. When a bar is popped, we calculate the area with the popped bar as smallest bar. How do we get left and right indexes of the popped bar – the current index tells us the 'right index' and index of previous item in stack is the 'left index'. Following is the complete algorithm.

1) Create an empty stack.

2) Start from first bar, and do following for every bar 'hist[i]' where 'i' varies from 0 to n-1. a) If stack is empty or hist[i] is higher than the bar at top of stack, then push 'i' to stack. b) If this bar is smaller than the top of stack, then keep removing the top of stack while top of the stack is greater. Let the removed bar be hist[tp]. Calculate

area of rectangle with hist[tp] as smallest bar. For hist[tp], the 'left index' is previous (previous to tp) item in stack and 'right index' is 'i' (current index).

3) If the stack is not empty, then one by one remove all bars from stack and do step 2.b for every removed bar.

6) آرش و دانیال در یک اتفاقی گیر افتاده اند که درش قفل است. قفل در به یک کامپیوتر متصل است که در هر بازه ای از زمان یک لینک لیست طولانی را به صورت رندوم تولید میکند که آنها میتوانند الگوریتمی را بر روی آن لوود کنند که با اجرای آن الگوریتم ریورس لیست داده شده را برگرداند، اگر پاسخ سریعاً برگردانده شود قفل باز میشود و در غیر اینصورت بمبی در اتاق است که منفجر میشود؛ از این رو آنها میخواهند الگوریتمشان در بهترین زمان ممکن پاسخ را برای کامپیوتر برگرداند تا نجات پیدا کنند. آنها از محدودیت حافظه برخورداری نیستند. باید سریعاً الگوریتمشان را طراحی کنند قبل از اینکه از گرسنگی تلف شوند. به آنها کمک کنید تا به بهترین جواب ممکن بتوانند برسند. توضیح دهید اگر لیست n عضوی باشد زمان اجرای این الگوریتم از اردر چند است و چگونه کار میکند. (20 نمره)

توجه: لیست یاد شده یک لیست پیوندی یک طرفه است.

توجه: مدت زمان پیش‌پردازش و پس‌پردازش مورد نظر نیست.

پاسخ:

نکته‌ی مهم قابل توجه در متن سوال "آنها از محدودیت حافظه برخوردار نیستند" است.

برای این کار با یک پیش‌پردازش از مرتبه‌ی $O(n)$ میتوان آدرس تمام نودها را بدست آورد و متناظر به هر کدام از آنها یک پوینتر اختصاص داد (آرایه‌ای از پوینتر ها) اکنون از $O(n)$ میتوان ریورس لیست داده شده را در خروجی برگرداند. اگر در هنگام نوع پیاده‌سازی ریورس کردن الگوریتمی بهینه‌تر استفاده شده باشد یا این مرحله را حذف کرده باشید نمره‌ی اضافه اختصاص میگیرد. برای مثال میتوان در همان پیش‌پردازش خانه‌ها را به لیست جدیدی مپ کرد یا از خاصیت پوینتر بهره گرفت اکنون برگرداندن لیست برعکس شده از $O(1)$ خواهد بود. ((توجه شود اردر پیش‌پردازش ملاک نبوده. درواقع درک استفاده از حافظه و منطق تریدآف میان حافظه و زمان مدنظر بوده و وابسته به توضیحات شما ممکن است نمره‌ی اضافی اختصاص یابد))

7) یک پشته در اختیار داریم که اعداد ۱ تا n به ترتیب در صف ورود به آن پشته هستند. (عدد ۱ پیش قدم شده است) هر بار با انجام عمل PUSH یک عنصر از صف به داخل استک اضافه میشود و هر بار با عمل POP یک عنصر از استک وارد صف خروج میشود. با ترکیب این دو عمل میتوانیم جایگشت‌های مختلفی از اعداد را در خروجی داشته باشیم. با توجه به این مطالب به سوالات زیر پاسخ دهید: (ورودی صف خروجی را سمت چپ آرایه در نظر بگیرید) (15 نمره)

الف) به ازای $n=10$ چگونه میتوانیم جایگشت $[4, 3, 2, 8, 7, 6, 9, 10, 1, 5]$ را داشته باشیم؟ توضیح دهید.

ابتدا تا عدد ۴ درون استک پوش میکنیم. سپس ۳ و ۲ را پاپ میکنیم. سپس تا عدد ۸ درون استک پوش کرده و بعد ۸ و ۷ و ۶ را پاپ میکنیم. ۹ را پوش و پاپ، سپس ۱۰ را پوش و پاپ میکنیم. در آخر هم هر چه در استک مانده را پاپ میکنیم.

ب) به ازای $n=10$ چگونه میتوانیم جایگشت $[4, 3, 8, 7, 9, 5, 10, 6, 2, 1]$ را داشته باشیم؟ توضیح دهید.

ساختن این صف امکان پذیر نیست. زیرا نمیتوان عدد ۶ در استک بالاتر از ۵ قرار دارد و نمیتوانیم آن را دیرتر از ۵ در صف خروجی داشته باشیم.

ج) توضیح دهید برای هر n ، همه جایگشت‌های خروجی دارای چه ویژگی‌ای هستند؟

هر بار که یک عنصر قرار است پاپ شود، تمام عناصری که در آن زمان در پشته قرار دارند باید به ترتیب نزولی پاپ شوند و در خروجی قرار گیرند. به عبارت دیگر در رشته خروجی، برای هر عدد n اعداد کوچک‌تر از آن باید تشکیل دنباله نزولی دهند.

8) در کتابخانه ای به 12 کتاب در سه قفسه وجود دارد که اندازه این کتاب ها را هر کدام با یک عدد مشخص کرده ایم.

12 : کتاب با بزرگترین اندازه

1: کتاب با کوچک ترین اندازه

مسئول کتابخانه می خواد تا کتاب ها را از چپ به راست در یک قفسه بچیند اما فضای زیادی برای پخش کردن کتاب ها ندارد و فقط می تواند کتاب ها را به صورت یک ستون از کتاب ها روی زمین و یا بین سه قفسه جا به جا کند.

نکته دیگر اینکه شما با کتاب های عتیقه سروکار دارید و یک کتاب با سایز بزرگتر نمی تواند به کتاب کوچک تر تکیه دهد (در قفسه ها کتاب بزرگتر نمی تواند در سمت راست کتاب کوچک تر قرار گیرد.) حال الگوریتمی ارائه دهید تا بتواند این کار را انجام دهد. دقت کنید امکان برداشتن چند کتاب وجود ندارد و در هر حرکت فقط راست ترین کتاب را به راست ترین موقعیت یک قفسه دیگر

ببرید (20 نمره)

[1]: 12 7 6 4 2 1

[2]: 9 5 3

[3]: 11 10 8

پاسخ:

این مسئله در واقع مسئله برج هانوی می باشد که به سادگی با استفاده از استک در اردر زمانی $O(2^n)$ قابل حل است. اما با این تفاوت که ما در برج های هانوی سه میله برای جا به جایی دیسک ها داشتیم اما در اینجا چهار میله تفاوت بعدی این است که سه تا از میله ها (قفسه ها) فعلی پر هستند. و نیاز است شما با میله 4 ام خالی ابتدا وضعیت را با توجه به حالت دلخواه مورد نیاز در بیاورید و سپس با استفاده از استک الگوریتم را پیش ببرید جهت دیدن کد این الگوریتم می توانید از این لینک استفاده کنید:

<https://www.geeksforgeeks.org/iterative-tower-of-hanoi/>

9) به کمک دو ساختمان داده صف و پشته الگوریتمی طراحی کنید که رشته ای از کاراکتر ها را دریافت و مشخص کند که این رشته متقارن است یا خیر. (15 نمره)

(رشته متقارن رشته ای است که از ابتدا به انتها خوانده شود یا از انتها به ابتدا یکسان باشد مثل MADAM)

پاسخ:

ابتدا تمام داده ها را وارد یک صف و یک پشته میتوان کرد. سپس شروع به پاپ کردن آنها میکنیم و در هر مرحله محتوای پاپ شده از استک و کیو را باهم مقایسه میکنیم. رویه ی یاد شده از پیچیدگی $O(n)$ تبعیت میکند. نکته: اگر تعداد حروف را بدانیم تنها با یک استک و $\theta(n)$ میتوانیم این کار را انجام دهیم که بهینه تر نیز خواهد بود.

10) لینک لیستی در اختیار داریم که ممکن است دوتا از عناصر آن به یکی از عناصر لینک لیست اشاره کنند اکنون با توجه به اینکه محدودیت حافظه داریم (در حد 5 خانه از حافظه در اختیارمون گذاشتن) و همچنین از سایز لیست هم اطلاعی نداریم بهترین الگوریتم ممکن برای اطلاع از این موضوع از چه اردری تبعیت میکند؟ روند کار الگوریتم مورد نظر را شرح دهید. (امتیازی) 20 نمره

نکته ۰: هر خانه از لیست نام برده شده دارای یک فیلد است که به خانه‌ی دیگری اشاره میکند و فیلد دیگری که متناظرا به هر خانه یک عدد یونیک انتساب شده است.

نکته ۱: تنها آدرس اولین خانه در دسترس ما قرار گرفته و برای دسترسی به دیگر خانه ها باید بر روی لیست پیمایش کنیم.

نکته ۲: از ترتیب خانه‌ها اطلاعی نداریم.

نکته ۳: از تعداد خانه‌ها نیز اطلاعی نداریم.

پاسخ:

مسئله همان مسئله‌ی تشخیص دور در لینکدلیست است. به دو اشاره‌گر نیازمندیم که یکی با سرعت کمتر و دیگری با سرعت بیشتر بر

روی لینکد لیست پیمایش کنند (به حرکت لاکپشت و خرگوش معروف می‌باشد)

اکنون هر بار که هر دو اشاره‌گر به یک خانه اشاره می‌کردند را می‌شماریم. اگر تعداد بازدید ها بیشتر از یک بار شد یعنی لینکد لیست

دور داره و لاغیر اگر اشاره‌گر خرگوش به ته لینکدلیست رسید یعنی دور نداره.

این کار از اردر $O(n)$ می‌باشد.

((به هر راه حل صحیح و هم اردری که محتوای لینکدلیست را خراب نکند نمره اختصاص داده می‌شود.))

پاینده و سربلند و اینها باشید :)