دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

درس ساختمان داده

دکتر مجتبی خلیلی

نمونه سوال از LinkedList, Queue, Stack

آبان ۱۴۰۲

آرش وشاق

۱ سوال۱

درستی یا نادرستی عبارات زیر را مشخص کنید. (برای تمام گزینه‌ها دلیل خود را بیان کنید)
الف) هزینه اضافه کردن یک عنصر جدید به یک لینکدلیست برابر با O(N) است.
ب) هزینه‌ی حذف آخرین عنصر از یک لیست یک‌سویه با دو اشاره‌گر f و r که به ترتیب به عناصر اول و آخر لیست اشاره می‌کنند از O(1) است.
ج) در یک لینکدلیست اگر یک اشاره‌گر به موقعیت n ام لیست را داشته باشیم، می‌توانیم عملیات درج یک عنصر جدید در موقعیت n را با اردر O(1) انجام دهیم.
د) در یک لیست حلقوی می‌توان با اردر O(1) به ابتدا و انتهای لیست افزود.

۲ سوال۲

رویه‌ی زیر به منظور وارون کردن یک لینکدلیست طراحی شده است. به این صورت که لینکدلیست L را به عنوان پارامتر می‌گیرد و لینکدلیست r که وارون آن است را برمی‌گرداند. در مورد درستی این روند توضیحی مختصر بنویسید.

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

شکل ۱:

۳ سوال۳

به سواالت زیر پاسخ دهید.
الف) نشان دهید که چطور با دو صف می‌توان یک استک ساخت. در مورد پیچیدگی زمانی عملیات استک توضیح دهید.
ب) نشان دهید که چطور با دو استک می‌توان یک صف ساخت. در مورد پیچیدگی زمانی عملیات صف توضیح دهید.

۴ سوال۴

یک استک در اختیار داریم که اعداد ۱ تا n به ترتیب در صف ورود به آن استک هستند. (عدد ۱ پیش قدم شده است.) یک صف خروجی هم داریم که در ابتدا خالی است. هر بار با انجام عمل push یک عنصر از صف ورودی به داخل استک اضافه می‌شود. هر بار با عمل pop یک عنصر از استک وارد صف خروجی می‌شود. با ترکیب این دو عمل می‌توانیم جایگشت‌های مختلفی از اعداد را در صف خروجی داشته باشیم. با توجه به این مطلب به سواالت زیر پاسخ دهید: (ورودی صف خروجی را سمت چپ آرایه در نظر بگیرید)

الف) به ازای n = 10 چگونه می‌توانیم جایگشت [1, 5, 10, 9, 6, 7, 8, 2, 3, 4] را در صف خروجی داشته باشیم؟
ب) به ازای n = 10 چگونه می‌توانیم جایگشت [1, 2, 6, 10, 5, 9, 7, 8, 3, 4] را در صف خروجی داشته باشیم؟
ج) برای هر n همه جایگشت‌های خروجی دارای چه ویژگی هستند؟

۵ سوال ۵

در کتابخانه‌ای ۱۲ کتاب در سه قفسه وجود دارند. اندازه این کتاب‌ها را هر کدام با یک عدد مشخص کرده‌ایم.

۱۲ : کتاب با بزرگترین اندازه

۱: کتاب با کوچک ترین اندازه

مسئول کتابخانه می‌خواهد کتاب‌ها را از چپ به راست در یک قفسه بچیند اما فضای زیادی برای پخش کردن کتاب‌ها ندارد و فقط می‌تواند کتاب‌ها را به صورت یک ستون از کتاب‌ها روی زمین و یا بین سه قفسه جا به جا کند. نکته دیگر این است که شما با کتاب‌های عتیقه سروکار دارید و یک کتاب با سایز بزرگتر نمی‌تواند به کتاب کوچکتر تکیه دهد (در قفسه‌ها کتاب بزرگتر نمی‌تواند در سمت راست کتاب کوچکتر 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 می‌توانید راست‌ترین کتاب را به راست‌ترین موقعیت یک قفسه دیگر ببرید.

[1]: 12 7 6 4 2 1

[2]: 9 5 3

[3]: 11 10 8

۶ سوال ۶

به کمک دو ساختمان داده صف و استک الگوریتمی طراحی کنید که رشته‌ای از کاراکترها را دریافت و مشخص کند که این رشته متفارن است یا خیر. رشته متقارن رشته‌ای است که از دو طرف یکسان خوانده شود. مثل madam

۷ سوال ۷

لینکدلیستی در اختیار داریم که ممکن است دوتا از عناصر آن به یکی از عناصر لینکدلیست اشاره کنند. اکنون با توجه به اینکه محدودیت حافظه داریم (در حد ۵ خانه از حافظه در اختیارمان گذاشته‌اند) 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.

نکته۱: هر خانه از لیست نام‌برده‌شده دارای یک فیلد است که به خانه دیگری اشاره می‌کند و فیلد دیگری که یک عدد یونیک برای آن نود است.

نکته۲: تنها آدرس اولین خانه در دسترس ما قرار گرفته و برای دسترسی به دیگر خانه‌ها باید بر روی لیست پیمایش کنیم.

نکته۳: از ترتیب خانه‌ها اطلاعی نداریم.

نکته۴: از تعداد خانه‌ها نیز اطلاعی نداریم.

۸ سوال ۸

		Array	Singly-Linked List	Doubly-Linked List
Access	by index	O(1)	O(N)	O(N)
Add	before first node	O(N)	O(1)	O(1)
	after given node	O(N)	O(1)	O(1)
	after last node	O(1)	O(N)	O(1)
Delete	the first node	O(N)	O(1)	O(1)
	a given node	O(N)	O(N)	O(1)
	the last node	O(1)	O(N)	O(1)
Search	a given node	O(N)	O(N)	O(N)

شکل ۲:

جدول بالا نشاندهنده اردر زمانی برخی عملیات بر روی آرایه و دو نوع لینکدلیست است. آیا این جدول مشکلی دارد؟ اگر بله، توضیح دهید.

۹ سوال ۹

کاربرد و طرز کار تابع foo را شرح دهید.

```
4   bool foo(string expr)
5   {
6       stack<char> temp;
7       for (int i = 0; i < expr.length(); i++) {
8           if (temp.empty()) {
9               temp.push(expr[i]);
10          }
11          else if ((temp.top() == '(' && expr[i] == ')')
12                  || (temp.top() == '{' && expr[i] == '}')
13                  || (temp.top() == '[' && expr[i] == ']')) {
14
15              temp.pop();
16          }
17          else {
18              temp.push(expr[i]);
19          }
20      }
21      if (temp.empty()) {
22          return true;
23      }
24      return false;
25  }
```

شکل ۳:

۱۰ سوال۱

درستی یا نادرستی عبارات زیر را مشخص کنید. (برای تمام گزینه‌ها دلیل خود را بیان کنید)
الف) هزینه اضافه کردن یک عنصر جدید به یک لینکدلیست برابر با O(N) است.
درست. بدترین حالت اضافه کردن یک نود به انتهای یک لینکدلیست است که برای این کار باید یک بار کامل بر روی لینکدلیست پیمایش کنیم.

ب) هزینه‌ی حذف آخرین عنصر از یک لیست یک‌سویه با دو اشاره‌گر 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 یک‌سویه نیازمند اشاره‌گر next نود قبلی هستیم لذا در یک لینکدلیست یکسویه ابتدا نیازمند یکبار پیمایش بر روی لینکدلیست خواهیم بود و دو اشاره‌گر مذکور تغییری در این روند ایجاد نمی‌کنند.

ج) در یک لینکدلیست اگر یک اشاره‌گر به موقعیت n ام لیست را داشته باشیم، می‌توانیم عملیات درج یک عنصر جدید در موقعیت n را با اردر O(1) انجام دهیم.
غلط. برای درج یک عنصر در موقعیت فعلی نیازمند اشاره‌گر next نود قبلی خواهیم بود که برای این کار بایستی یک بار از ابتدا تا یک خانه عقبتر از موقعیت فعلی را پیمایش کنیم. این کار از اردر O(n) است.

د) در یک لیست حلقوی می‌توان با اردر O(1) به ابتدا و انتهای لیست افزود.
پاسخ این سوال بستگی به فرضیات شما دارد. اگر لینکدلیست را یکطرفه درنظر بگیریم پاسخ مسئله غلط و اگر دوطرفه در نظر بگیریم پاسخ مسئله صحیح است.

۱۱ سوال۲

رویه‌ی زیر به منظور وارون کردن یک لینکدلیست طراحی شده است. به این صورت که لینکدلیست L را به عنوان پارامتر می‌گیرد و لینکدلیست r که وارون آن است را برمی‌گرداند. در مورد درستی این روند توضیحی مختصر بنویسید.

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

شکل ۴:

با کمی دقت می‌توان یافت که در رویه بالا در آخرین باری که تابع فراخوانی می‌شود به دلیل اینکه شرط L = null برقرار است دستور return L یا در واقع همان return null اجرا می‌شود. این مقدار در اولین خانه لینکدلیست r ذخیره می‌شود بنابراین رویه داده‌شده درست کار نمی‌کند.

۱۲ سوال ۳

به سواالت زیر پاسخ دهید.

الف) نشان دهید که چطور با دو صف می‌توان یک استک ساخت. در مورد پیچیدگی زمانی عملیات استک توضیح دهید.

هنگام افزودن عناصر جدید، آنها را به یک صف اضافه می‌کنیم که این عمل از اردر O(1) است. برای پیاده سازی عملگر pop اعضای موجود در این صف را dequeue می‌کنیم و به صف دوم اضافه می‌کنیم (اردر O(n)) تا به آخرین عضو در صف اول برسیم. این عضو باقی‌مانده را برمی‌گردانیم.

ب) نشان دهید که چطور با دو استک می‌توان یک صف ساخت. در مورد پیچیدگی زمانی عملیات صف توضیح دهید.

هنگام افزودن عناصر آنها را به استک اول اضافه می‌کنیم. (اردر O(1)) برای پیاده‌سازی عملگر dequeue از استک دوم یک عنصر pop می‌کنیم. اگر استک دوم خالی بود، ابتدا همه اعضای استک اول را pop می‌کنیم و به استک دوم اضافه می‌کنیم. (اردر O(n)) سپس یک عضو از استک دوم pop می‌کنیم.

۱۳ سوال ۴

یک استک در اختیار داریم که اعداد ۱ تا n به ترتیب در صف ورود به آن استک هستند. (عدد ۱ پیش قدم شده است.) یک صف خروجی هم داریم که در ابتدا خالی است. هر بار با انجام عمل push یک عنصر از صف ورودی به داخل استک اضافه می‌شود. هر بار با عمل pop یک عنصر از استک وارد صف خروجی می‌شود. با ترکیب این دو عمل می‌توانیم جایگشت‌های مختلفی از اعداد را در صف خروجی داشته باشیم. با توجه به این مطلب به سواالت زیر پاسخ دهید: (ورودی صف خروجی را سمت چپ آرایه در نظر بگیرید)

الف) به ازای n = 10 چگونه می‌توانیم جایگشت [1, 5, 10, 9, 6, 7, 8, 2, 3, 4] را در صف خروجی داشته باشیم؟

ابتدا تا عدد ۴ درون استک push می‌کنیم. سپس ۴ و ۳ و ۲ را pop می‌کنیم. سپس تا عدد ۸ درون استک push می‌کنیم و بعد ۸ و ۷ و ۶ را pop می‌کنیم. ۹ را push و pop سپس ۱۰ را push و pop می‌کنیم و در آخر هر چه در استک مانده را pop می‌کنیم.

ب) به ازای n = 10 چگونه می‌توانیم جایگشت [1, 2, 6, 10, 5, 9, 7, 8, 3, 4] را در صف خروجی داشته باشیم؟

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

ج) برای هر n همه جایگشت‌های خروجی دارای چه ویژگی هستند؟

هر بار که یک عنصر قرار است پاپ شود، تمام عناصری که در آن زمان در استک 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.

۱۴ سوال ۵

در کتابخانه‌ای ۱۲ کتاب در سه قفسه وجود دارند. اندازه این کتاب‌ها را هر کدام با یک عدد مشخص کرده‌ایم.

۱۲ : کتاب با بزرگترین اندازه

۱: کتاب با کوچک ترین اندازه

مسئول کتابخانه می‌خواهد کتاب‌ها را از چپ به راست در یک قفسه بچیند اما فضای زیادی برای پخش کردن کتاب‌ها ندارد و فقط می‌تواند کتاب‌ها را به صورت یک ستون از کتاب‌ها روی زمین و یا بین سه قفسه جا به جا کند. نکته دیگر این است که شما با کتاب‌های عتیقه سروکار دارید و یک کتاب با سایز بزرگتر نمی‌تواند به کتاب کوچکتر تکیه دهد (در قفسه‌ها کتاب بزرگتر نمی‌تواند در سمت راست کتاب کوچکتر 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 می‌توانید راست‌ترین کتاب را به راست‌ترین موقعیت یک قفسه دیگر ببرید.

[1]: 12 7 6 4 2 1

[2]: 9 5 3

[3]: 11 10 8

این مسئله در واقع مسئله برج هانوی می‌باشد که به سادگی با استفاده از استک در اردر زمانی (۲ به توان n) قابل حل است اما با این تفاوت که ما در برج‌های هانوی سه میله برای جا به جایی دیسک‌ها داشتیم اما در اینجا چهار میله. تفاوت بعدی این است که سه تا از میله‌ها (قفسه‌های) فعلی پر هستند و نیاز است که شما با میله چهارم خالی ابتدا وضعیت را با توجه به حالت دلخواه مورد نیاز در بیاورید و سپس با استفاده از استک الگوریتم را پیش ببرید.

See here

۱۵ سوال ۶

به کمک دو ساختمان داده صف و استک الگوریتمی طراحی کنید که رشته‌ای از کاراکترها را دریافت و مشخص کند که این رشته متفارن است یا خیر. رشته متقارن رشته‌ای است که از دو طرف یکسان خوانده شود. مثل madam

ابتدا تمام داده‌ها را وارد یک صف و یک استک می‌کنیم. سپس شروع به پاپ کردن آنها می‌کنیم و در هر مرحله محتوای پاپ شده از استک و صف را با هم مقایسه می‌کنیم. رویه‌ی یاد شده از پیچیدگی O(n) تبعیت می‌کند.

۱۶ سوال ۷

لینکدلیستی در اختیار داریم که ممکن است دوتا از عناصر آن به یکی از عناصر لینکدلیست اشاره کنند. اکنون با توجه به اینکه محدودیت حافظه داریم (در حد ۵ خانه از حافظه در اختیارمان گذاشته‌اند) 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.

نکته۱: هر خانه از لیست نام‌برده‌شده دارای یک فیلد است که به خانه دیگری اشاره می‌کند و فیلد دیگری که یک عدد یونیک برای آن نود است.

نکته۲: تنها آدرس اولین خانه در دسترس ما قرار گرفته و برای دسترسی به دیگر خانه‌ها باید بر روی لیست پیمایش کنیم.

نکته۳: از ترتیب خانه‌ها اطلاعی نداریم.

نکته۴: از تعداد خانه‌ها نیز اطلاعی نداریم.

See here

۱۷ سوال ۸

		Array	Singly-Linked List	Doubly-Linked List
Access	by index	O(1)	O(N)	O(N)
Add	before first node	O(N)	O(1)	O(1)
	after given node	O(N)	O(1)	O(1)
	after last node	O(1)	O(N)	O(1)
Delete	the first node	O(N)	O(1)	O(1)
	a given node	O(N)	O(N)	O(1)
	the last node	O(1)	O(N)	O(1)
Search	a given node	O(N)	O(N)	O(N)

شکل ۵:

جدول بالا نشاندهنده اردر زمانی برخی عملیات بر روی آرایه و دو نوع لینکدلیست است. آیا این جدول مشکلی دارد؟ اگر بله، توضیح دهید.

خیر - اطلاعات جدول صحیح است.

See here

۱۸ سوال ۹

کاربرد و طرز کار تابع foo را شرح دهید.

```
4   bool foo(string expr)
5   {
6       stack<char> temp;
7       for (int i = 0; i < expr.length(); i++) {
8           if (temp.empty()) {
9               temp.push(expr[i]);
10          }
11          else if ((temp.top() == '(' && expr[i] == ')')
12                  || (temp.top() == '{' && expr[i] == '}')
13                  || (temp.top() == '[' && expr[i] == ']')) {
14
15              temp.pop();
16          }
17          else {
18              temp.push(expr[i]);
19          }
20      }
21      if (temp.empty()) {
22          return true;
23      }
24      return false;
25  }
```

شکل ۶:

See here