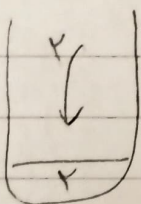


۱ جواب سوال ۱:

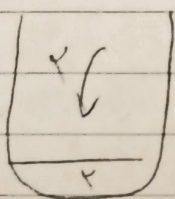
۲ الف)

۳ از دو استک برای ذخیره سازی میزنیم استفاده می کنیم که بدین صورت
۴ که در مرحله push کردن عنصر ها در استک دوم آن (به صورت نزولی
۵ ذخیره می کنیم) بالاترین عنصر کمینه است. ۱۰.

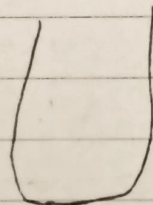
۵/۱ و ۴/۱ و ۲



Stack 1

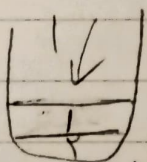


Stack 2

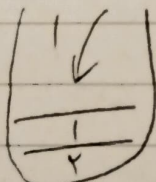


Stack 3

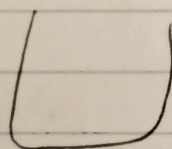
چون یک از دو کوچکتر است آن را به طور مادی در Stack 2 push می کنیم.



Stack 1

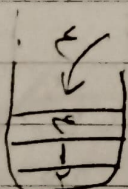


Stack 2

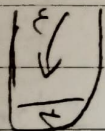


Stack 3

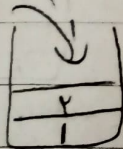
چون ۴ از head Stack 2 که حتماً تا جایی که ۴ از عنصرهای
استک ۲ کوچکتر باشد و یا استک ۲ خالی نشده باشد که دونه دونه
از استک ۲ pop می کنیم و سپس عنصر ۴ را در استک ۲ push می کنیم
و بقیه را از استک ۳ بیرون می آوریم.



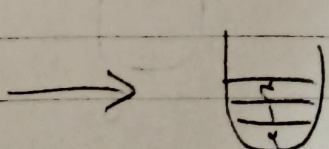
Stack 1



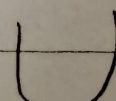
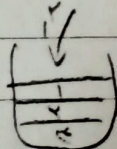
Stack 2



Stack 3



Stack 1

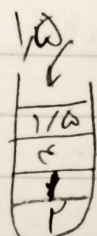


Stack 3

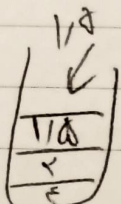
ادامه جواب سوال ۱:

ادامه الف)

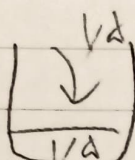
حالا ۵ را به همان کس اضافه می کنیم.



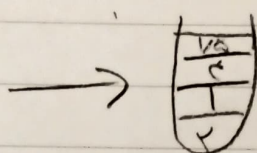
Stack 1



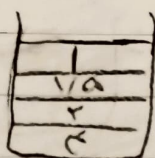
Stack 2



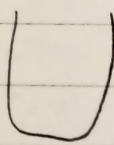
Stack 3



Stack 1



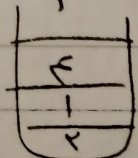
Stack 2



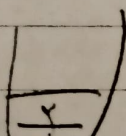
Stack 3

حالا برای گرفتن که چقدر عنصر بالاترین عنصر Stack 2 را pop می کنیم.
تعریف بالا insert بود که حالا delete را تعریف می کنیم.
بالاترین عنصر Stack 1 را pop می کنیم و در استک دوم تاجایی که
به آن عنصر pop شده نرسیدیم عناصر Stack 2 را pop می کنیم و در
Stack 3 push می کنیم. حالا بعد از حذف عنصر مورد نظر که عناصر را
از Stack 3 به استک دوم برمی گردانیم.

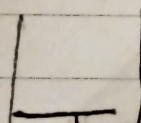
۱, ۵



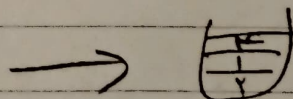
Stack 1



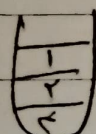
Stack 2



Stack 3



Stack 1



Stack 2

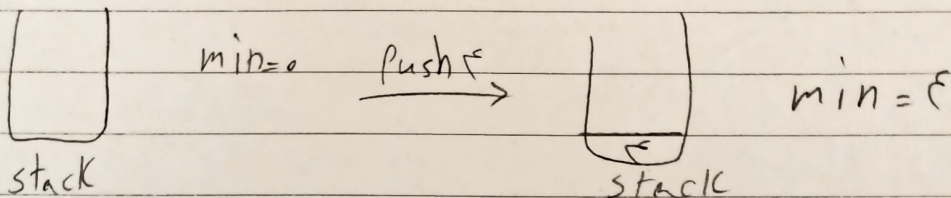


Stack 3

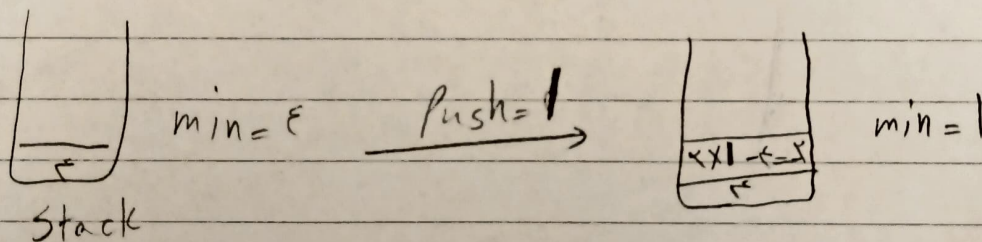
۱. ارائه جواب سوال ۱

(ب)

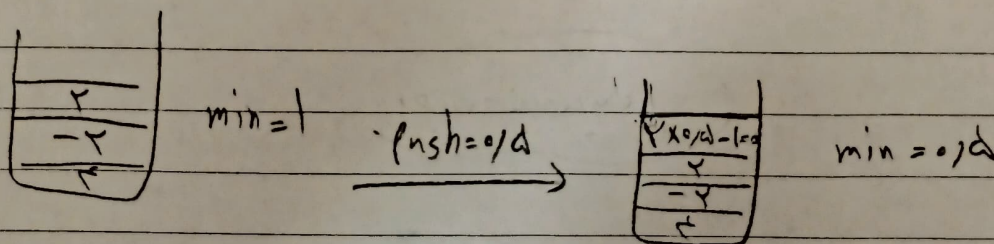
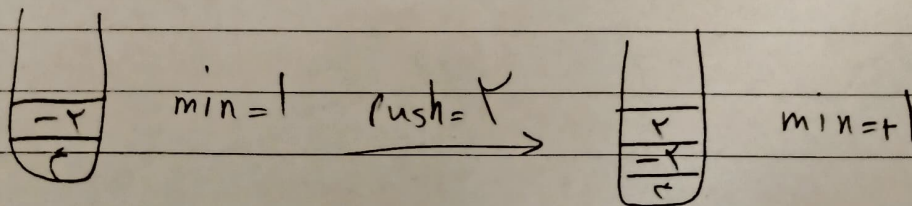
متغیری به نام min داریم که در اول که استک خالی است برای push کردن مقدار جدید min را برابر همان مقدار قرار می دهیم.



در ادامه برای اضافه کردن عنصر جدید جب می کنیم و اگر مقدار جدید از top استک کمتر باشد در استک مقدار $2 \times \min$ را ذخیره می کنیم و min را برابر ورودی جدید قرار می دهیم.



و اگر مقدار ورودی جدید از top استک بیشتر بود همان را push می کنیم.



اداره جواب سوال ۱۰۰

اداره ب ()

برای pop کردن عنصر بالای استک را بهی داریم اگر
از min کمتر بود که min را برابر 2 تا min قبلی به علاوه top استک
می داریم و مقدار بهگشتی از تابع pop را برابر $\frac{top + min}{2}$ قرار
می دهیم.

۵
۲
-۲
۴

$$min = 0/5 \quad pop = 0 \rightarrow$$

۲
-۲
۴

$$min = 2 \times 0/5 - 0/5 = 1$$

$$min = 1$$

$$خوبی = \frac{0+1}{2} = 0/5$$

در غیر این صورت بالاترین را بهی گردونیم.

۲
-۲
۴

$$min = 1 \quad pop = 2 \rightarrow$$

-۲
۴

$$min = 1$$

۱ جواب سوال ۲

۲ الف) دو سنجش برای $index1$ و $index2$ می‌داریم
 ۳ که اول برابر ۰ و دومی برابر $n-1$ هست. برای هر کدام
 ۴ از استک‌های ۱ و ۲ تابع $push1$ و $push2$ را انکاره تعریف می‌کنیم

۵ $index1 = 0$ و $index2 = n-1$;

۶ $push1(x)$

۷ $if (index2 - index1 == 1)$

۸ $return "Array is full";$

۹ $Array[index1] = x;$

۱۰ $index1 = index1 + 1;$

۱۱ $push2(x)$

۱۲ $if (index2 - index1 == 1)$

۱۳ $return "Array is full";$

۱۴ $Array[index2] = x;$

۱۵ $index2 = index2 - 1;$

۱۶ $pop1()$

۱۷ $if (index1 == 0)$

۱۸ $return "Empty";$

۱۹ $index1 = index1 - 1;$

۲۰ $return Array[index1 + 1];$

۲۱ $pop2()$

۲۲ $if (index2 == n-1)$

۲۳ $return "Empty";$

۲۴ $index2 = index2 + 1;$

۲۵ $return Array[index2 + 1];$

۱ ادامه جواب سوال ۲:

- ۲ (ب) سه متغیر $index1$, $start2$, $end2$, $index3$ را برابر ۰ و $[n/4]$ و $[n/2]$ و $n-1$ قرار می دهیم و برای $push$ کردن در استک اول $index1$ را یکی جلوی می بینیم و برای $push$ کردن در استک دوم $end2$ را یکی جلوی می بینیم و برای استک سوم $index3$ را یکی کمر می کنیم و متادیر را در آن خانه ها می ریزیم.

- ۸ اگر $index1$ به $start2$ رسید و $end2$ به $index3$ نرسیده بود می توان
۹ تمامی متادیر استک دوم را یکی به جلو شیفت بدهیم تا بتوان
۱۰ متادیر بیشتری برای استک اول در نظر گرفت در غیر این صورت
۱۱ آرایه پر شده است.

- ۱۲ همچنین اگر $index3$ به $end2$ رسید و از آن در برای استک
۱۳ دوم (پشتین) خالی بود که آنگاه استک دوم را یکی به عقب می بینیم
۱۴ در غیر این صورت آرایه پر شده است.

- ۱۵ برای pop کردن $index1$ و $end2$ را کمر می کنیم و از آزادی کنیم.
۱۶ (ج) سه متغیر $index1$ و $index2$ و $index3$ را برابر ۰ و $[n/3]$ و $[2n/3]$ قرار می دهیم و طول هر استک $[n/3]$ می شود آنگاه
۱۷ برای $push$ کردن متادیر در هر استک خانه $index$ را برابر مقدار قرار می دهیم
۱۸ و سپس یکی به $index$ اضافه می کنیم و برای pop یکی کمر می کنیم متادیر
۱۹ نباید به مرز قبلی تجاوز کند.

$push1(x)$

```
if (index1 + 1 == [n/4])
    return "Full";
Array[index1] = x;
index1 ++;
```

$push2(x)$

```
if (index2 + 1 == [2n/3])
    return "Full";
Array[index2] = x;
index2 ++;
```

$push3(x)$

```
if (index3 + 1 == n)
    return "Full";
Array[index3] = x;
index3 ++;
```

$pop1()$

```
if (index1 == 0)
    return "Empty";
index1 --;
return Array[index1 + 1];
```


۱ جواب سوال ۳

از صف برای این سوال کپی می‌گیریم، به این صورت که اعضای
هر سطح را به صف اضافه می‌کنیم سپس به هر کدام از آن رسیدیم بچه‌آن را
به صف اضافه می‌کنیم سپس خود آن را از صف خارج می‌کنیم و به result
اضافه می‌کنیم.

Q = Queue();

resultNode = null;

current = resultNode; current2 = head -> next;

Q.add(head);

while (!Q.isEmpty()) {

while (current2 != null) {

Q.add(current2);

current2 = current2 -> next;

}

current -> next = Q.remove();

current = current.next;

current2 = current.child;

}

حال resultNode کپی Node هست که جواب مورد نظر ما هست.

Subject:

Year. Month. Date.

جواب سوال ۴ :

۱ دو متغیر current و current2 داریم که ابتدا هر دو
 ۲ برابر head هستند و current یکی یکی جلو می رود در حالی که
 ۳ current2 دو تا دو تا جلو می رود. اگر این دو به هم برسند آننگاه
 ۴ به این معنی است که دور وجود دارد در غیر این صورت دور
 ۵ وجود ندارد.

۶
 ۷ $current = head$
 ۸ $current2 = head$

۹
 ۱۰ $while (True) \{$

۱۱
 ۱۲ $if (current2 == null || current2 \rightarrow next == null)$

۱۳ $return True;$

۱۴ $else if (current2 == current)$

۱۵ $return False;$

۱۶ $current = current \rightarrow next;$

۱۷ $current2 = current2 \rightarrow next \rightarrow next;$

۱۸ $\}$

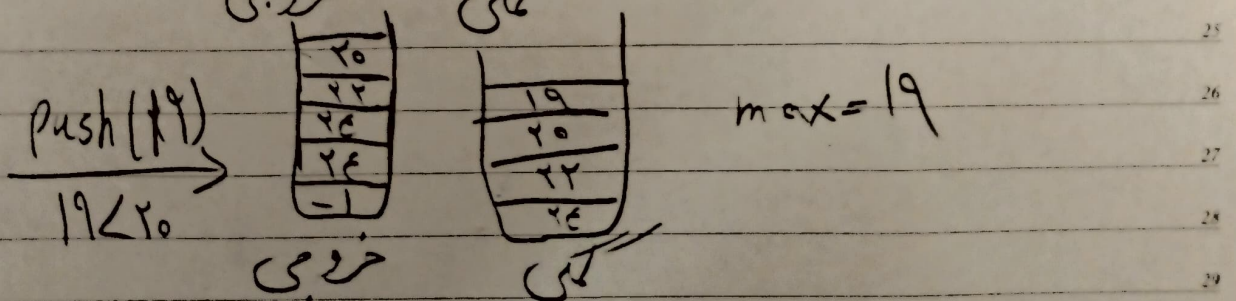
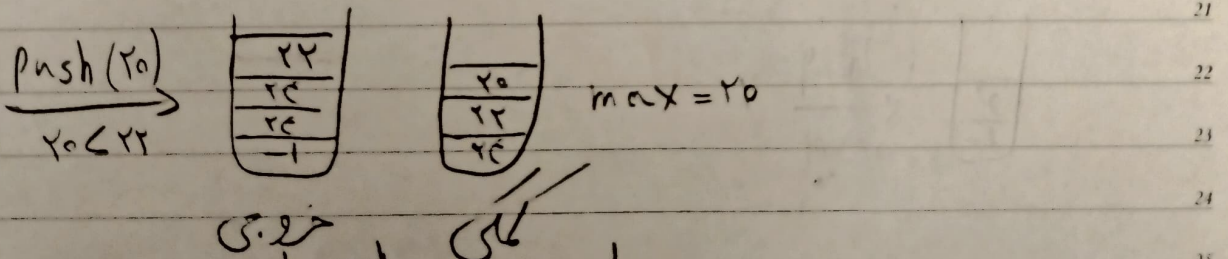
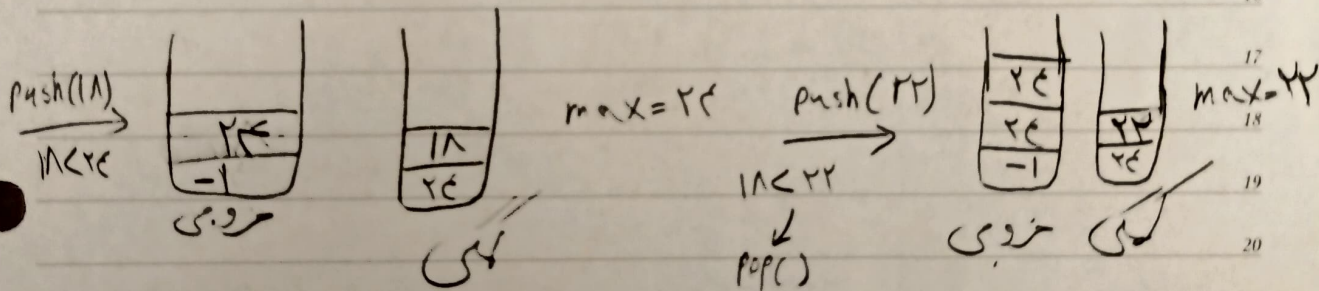
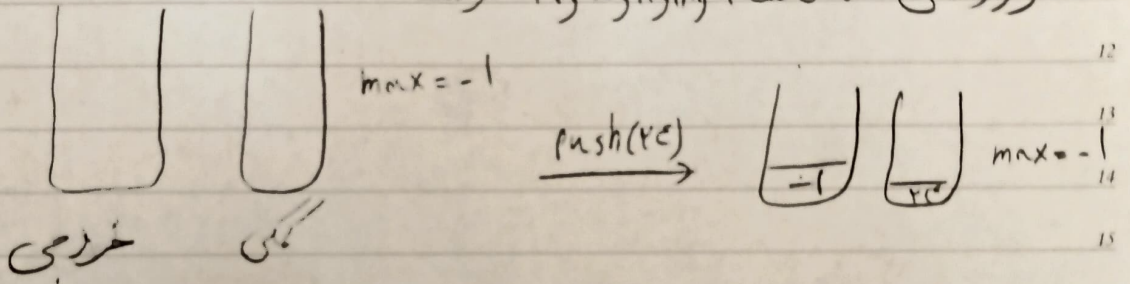
۱۹
 ۲۰
 ۲۱ حال اگر خروجی تابع False باشد بدین معنا

۲۲
 ۲۳ است که دور وجود دارد.

جواب سوال ۵

از ته صف شروع می کنیم و در استک خروجی یک صفی یک push می کنیم حالا یک استک داریم که به ما کمک می کند برای خروجی در هر مرحله عنصری که در صف هستیم را می خوانیم اگر از اول استک کمکی کوچکتر بود آن را در استک کمکی پویش می کنیم و مقدار متغیر max را که آن متغیر بزرگتر است در استک خروجی پویش می کنیم، در غیر این صورت تاجایی که بتوان از استک خروجی pop می کنیم.

متغیری به نام max داریم که در اول ۱- هست. اگر استک کمکی خالی بود آنگاه max را منفی ۱ می گذاریم. ورودی: ۵، ۲، ۱۸، ۲۲، ۲۰، ۱۹ صرف



Subject:

Year. Month. Date.

۱. نام جواب سوال ۵:

۲. روش کار به این صورت است که ابتدا متغیر برابر ۱ - است
۳. اسم متغیر را max می‌کنیم.
۴. از آخر صف شروع می‌کنیم و تک تک به عقب می‌رویم تا به اول
۵. صف برسیم. دو استک در اختیار داریم با نام های خروجی و کمکی.
۶. در هر مرحله در عنصر صف را با عنصر top استک کمکی مقایسه می‌کنیم
۷. اگر از آن کوچکتر بود آن را در استک کمکی push می‌کنیم، در غیر این صورت
۸. تا جایی که عنصر بزرگتر شده از top استک کمکی بزرگتر مساوی باشد
۹. از استک کمکی pop می‌کنیم و در نهایت آن را به استک کمکی خالی برمی‌گردانیم
۱۰. و با عنصری داخل آن که از عنصر ورودی بزرگتر است push
۱۱. می‌کنیم. مقدار max را اگر استک کمکی خالی شد برابر ۱ - قرار
۱۲. می‌دهیم در غیر این صورت برابر top استک کمکی قرار می‌دهیم.
۱۳. در هر مرحله قبل از تغییر دادن max آن را در استک خروجی push
۱۴. می‌کنیم.

15

16

17

18

19

20

21

22

23

24

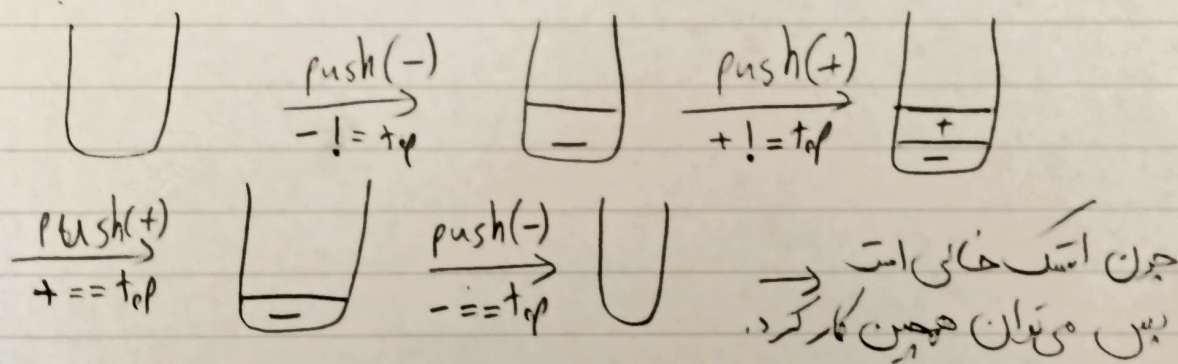
25

26

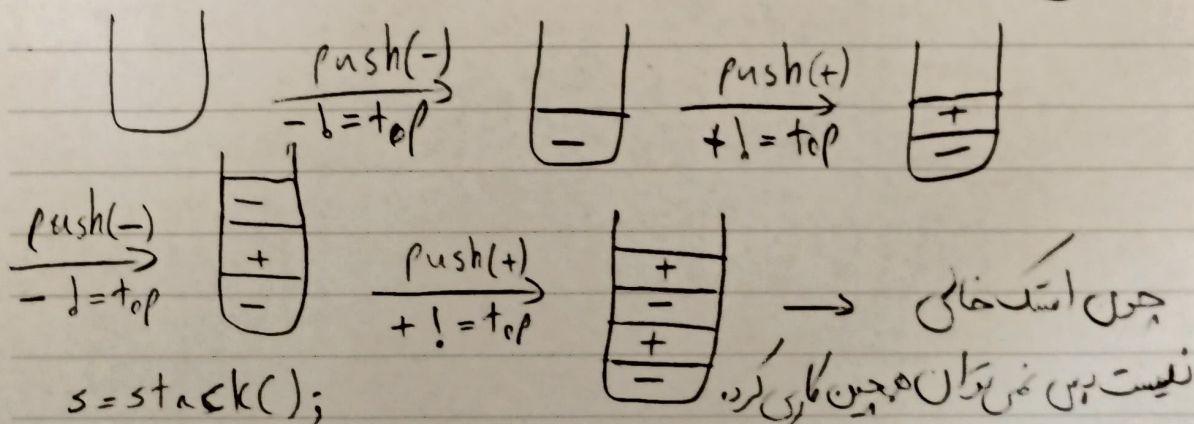
جواب سوال ۴ :

برای حل این سوال مانند انترگذاری صحیح عمل می کنیم این گونه که از یک استک استفاده می کنیم ، از اول ورودی شروع می کنیم و به ترتیب عناصر را در استک پوش می کنیم ، اگر منتهی ورودی برابر با top است یکی برد میسر pop استگ را pop می کنیم در غیر این صورت آن را در استک $push$ می کنیم ، اگر در آخر ورودی ، استک خالی بود پس شرایط ورودی نیاز سوال فراهم است و می توان این کار را انجام داد که در زیر این صورت همچنین کاری غیر ممکن است.

ورودی ۴ : - + + -



ورودی ۵ : - + - +



```
s = stack();
```

```
for xin input: if x == s.top: s.pop(); else s.push(x);
```

```
if x.isEmpty(): return True; else: return False;
```

اگر True برگرداند پس مسئله قابل حل است.