حل تمرین سری دوم(آرایه و پشته)

ساختمان داده ها و الگوریتم

سجاد هاشمیان، بهار ۱۴۰۰

سوال ۱.

ارشد ۹۷

در یک آرایه عددی A به طول m؛ برخی از عناصر ۱ تا n که $m < n$ است، ظاهر شده‌اند. بهترین الگوریتم برای یافتن بزرگ‌ترین عنصری که در آرایه ظاهر نشده است، دارای چه مرتبه زمانی است؟

۱) $O(n)$

۲) $O(m)$

۳) $(n \log n)$

۴) $O(m \log m)$

رویکرد سطلی

مشاهده‌ای دقیق تر

- اگر لیست اعداد ما مرتب بود، چگونه عمل می‌کردیم؟

رویکرد سطلی

مشاهده‌ای دقیق تر

- اگر لیست اعداد ما مرتب بود، چگونه عمل می‌کردیم؟
- بزرگ‌ترین عنصر موجود را داشتیم.

```
if(A[m-1]<n):
    print(n)
else:
    for i in range(m-1,1,-1):
        if(A[i]-1!=A[i-1]):
            print(A[i]-1)
```

$\longrightarrow O(m)$

رویکرد سطلی

مشاهده‌ای دقیق تر

- اگر لیست اعداد ما مرتب بود، چگونه عمل می‌کردیم؟
- بزرگ‌ترین عنصر موجود را داشتیم.

```
if(A[m-1]<n):
    print(n)
else:
    for i in range(m-1,1,-1):
        if(A[i]-1!=A[i-1]):
            print(A[i]-1)
```

- مشکل؟ بهترین مرتب سازی برمبنای مقایسه از مرتبه $O(m \log m)$ است.

رویکرد سطلی

مشاهده‌ای دقیق تر

- حالا که لیست مرتب نیست، چه چیزی کار را خراب می‌کند؟
- نابه‌جایی‌ها: عدد iام قبل از یک عدد j آمده که j<i هست.
- بررسی سطلی: عدد iام سر جاش نیست!

رویکرد سطلی

مگو این سخن جز مر اهل بیان را.

ناصر خسرو

- خب پس بیایم عدد i رو ببریم سر جاش!

```
def count(A,m,n):
    t=[0]*(n+1)
    for i in A:
        t[i]+=1
    for i in range(n,0,-1):
        if(t[i]==0):
            return i
```

$\left.\begin{matrix}\\\\\\\end{matrix}\right\} \longrightarrow O(m)$

$\left.\begin{matrix}\\\\\\\end{matrix}\right\} \longrightarrow ?$

رویکرد سطلی

گزینه ۲.

- خب پس بیایم عدد i رو ببریم سر جاش!

```
def count(A,m,n):
t=[0]*(n+1)
for i in A:                    } ⟶ O(m)
    t[i]+=1
for i in range(n,0,-1):        }
    if(t[i]==0):               } ⟶ O(m)
        return i
```

سوال ۲.

ارشد ۹۶

می‌خواهیم در یک آرایه n تایی نامرتب، عنصری که حداقل $1 + \lfloor \frac{n}{2} \rfloor$ بار تکرار شده است را بیابیم؛ کدام گزینه صحیح است؟

۱) الگوریتمی با هزینه حداکثر $O(n)$ وجود دارد.

۲) الگوریتمی با هزینه حداکثر $O(\sqrt{n})$ وجود دارد.

۳) بهترین الگوریتم با زمان اجرای میانگین $O(n)$ و مبتنی بر درهم‌سازی است.

۴) بهترین الگوریتم با زمان اجرای میانگین $O(n \log n)$ و مبتنی بر مرتب‌سازی است.

رای اکثریت

Majority Vote Problem

- راه حل مقدماتی: $O(n^2)$
- درخت جستجو دودویی متوازن: $O(n \log n)$
- شمارش به وسیله توابع Hash: $T(Cn) = O(n)$
  - که C می‌تواند نسبت به n نجومی باشد!
- الگوریتم Boyer-Moore: $O(n)$
- جستجو تصادفی: $O(\sqrt{n})$
  - پاردوکس تولد
  - کلاس محاسباتی BPP، ZPP و P ؟

رای اکثریت

Majority Vote Problem

- راه حل مقدماتی: $O(n^2)$
- درخت جستجو دودویی متوازن: $O(n \log n)$

```
for a in A:
    t=0
    for i in (0,n):
        if(A[i]==a):
            t+=1
    if(t>n/2):
        return a
```

- شمارش به وسیله توابع sh
  - که C می‌تواند نسبت به n نج
- الگوریتم Boyer-Moore:
- جستجو تصادفی:
  - پاردوکس تولد
  - کلاس محاسباتی BPP، ZPP و P ؟

رای اکثریت

Majority Vote Problem

- راه حل مقدماتی: $O(n^2)$
- درخت جستجو دودویی متوازن: $O(n \log n)$
- شمارش به وسیله توابع Hash: $T(Cn) = O(n)$
  - که C می‌تواند نسبت به n نجومی باشد!
- الگوریتم Boyer-Moore: $O(n)$ ← گزینه ۱
- جستجو تصادفی: $O(\sqrt{n})$
  - پاردوکس تولد
  - کلاس محاسباتی BPP، ZPP و P ؟

الگوریتم Boyer-Moore

شرح الگوریتم

```
def boyer_moore(A):
    m = -1
    t = 0
    for a in A:
        if(t == 0):
            m = a
        if(a == m):
            t += 1
        if(a != m):
            t -= 1
    return m
```

الگوریتم Boyer-Moore

بررسی یک مثال

```
list A= [1, 4, 3, 1, 1, 1, 2, 5, 1]
initial value m, t= -1, 0
for 1 : m, t= 1 ,  1
for 4 : m, t= 1 ,  0
for 3 : m, t= 3 ,  1
for 1 : m, t= 3 ,  0
for 1 : m, t= 1 ,  1
for 1 : m, t= 1 ,  2
for 2 : m, t= 1 ,  1
for 5 : m, t= 1 ,  0
for 1 : m, t= 1 ,  1
result = 1
```

```
def boyer_moore(A):
    m = -1
    t = 0
    for a in A:
        if(t == 0):
            m = a
        if(a == m):
            t += 1
        if(a != m):
            t -= 1
    return m
```

الگوریتم Boyer-Moore

اثبات درستی

اگر یک عنصر اکثریت وجود داشته باشد ، الگوریتم همیشه آن را پیدا می کند. زیرا ، با فرض اینکه عنصر اکثریت m باشد ، بگذارید c عددی باشد که در هر مرحله از الگوریتم تعریف شده باشد یا در صورت عدم وجود عنصر ذخیره شده در متر ، یا در صورت غیر منفی بودن شمارنده باشد. سپس در هر مرحله که الگوریتم با مقداری برابر با m روبرو می شود ، مقدار c یک افزایش می یابد و در هر مرحله که با مقدار متفاوتی روبرو می شود ، ممکن است مقدار c یک افزایش یا کاهش یابد. اگر m واقعاً اکثریت باشد ، بیشتر از کاهش خواهد بود و c در پایان الگوریتم مثبت خواهد بود. اما این فقط وقتی درست است که عنصر ذخیره شده نهایی m باشد ، عنصر اکثریت.

سوال ۳.

ارشد ۹۵

میخواهیم برای عدد صحیح و مثبت n بزرگ‌ترین عدد صحیح و مثبت x را پیدا کنیم که $x^2 \leq n$ باشد.
بهترین الگوریتم برای یافتن چنین عددی از چه مرتبه زمانی است؟

۱) $O(n)$

۲) $O(\sqrt{n})$

۳) $O(\log n)$

۴) $O(n \log n)$

سوال ۳.

گزینه ۳.

میخواهیم برای عدد صحیح و مثبت n بزرگ‌ترین عدد صحیح و مثبت x را پیدا کنیم که $x^2 \leq n$ باشد.
بهترین الگوریتم برای یافتن چنین عددی از چه مرتبه زمانی است؟

- کافیست تا x را در لیست اعداد صحیح $[1 \cdots n]$ جستجو کنیم، از آنجا که $x^2 \leq n$ پس حتما x در این لیست موجود است، برای جستجو نیز کافیست تا از یک باینری سرچ استفاده کنیم.

سوال ۳.

گزینه ۳.

میخواهیم برای عدد صحیح و مثبت n بزرگ‌ترین عدد صحیح و مثبت x را پیدا کنیم که $x^2 \leq n$ باشد.
بهترین الگوریتم برای یافتن چنین عددی از چه مرتبه زمانی است؟

- کافیست تا x را در لیست اعداد صحیح $[1 \cdots n]$ جستجو کنیم، از آنجا که $x^2 \leq n$ پس حتما x در این لیست موجود است، برای جستجو نیز کافیست تا از یک باینری سرچ استفاده کنیم.

- BigNum: البته می‌توان از رابطه $x = \lfloor \sqrt{n} \rfloor$ استفاده کرد. (مرتبه زمانی؟)

سوال ۴.

ارشد ۹۵

اگر الگوریتم جستجوی دودویی را برای جستجوی عناصر آرایه $A[1..9] = [-1,2,10,20,25,29,35,45,50]$ به کار ببریم، میانگین تعداد مقایسه‌ها برای جستجوی موفق و ناموفق، به طور تقریبی چقدر است؟

۱) موفق ۲,۸ ناموفق ۳,۸

۲) موفق ۲,۸ ناموفق ۲,۸

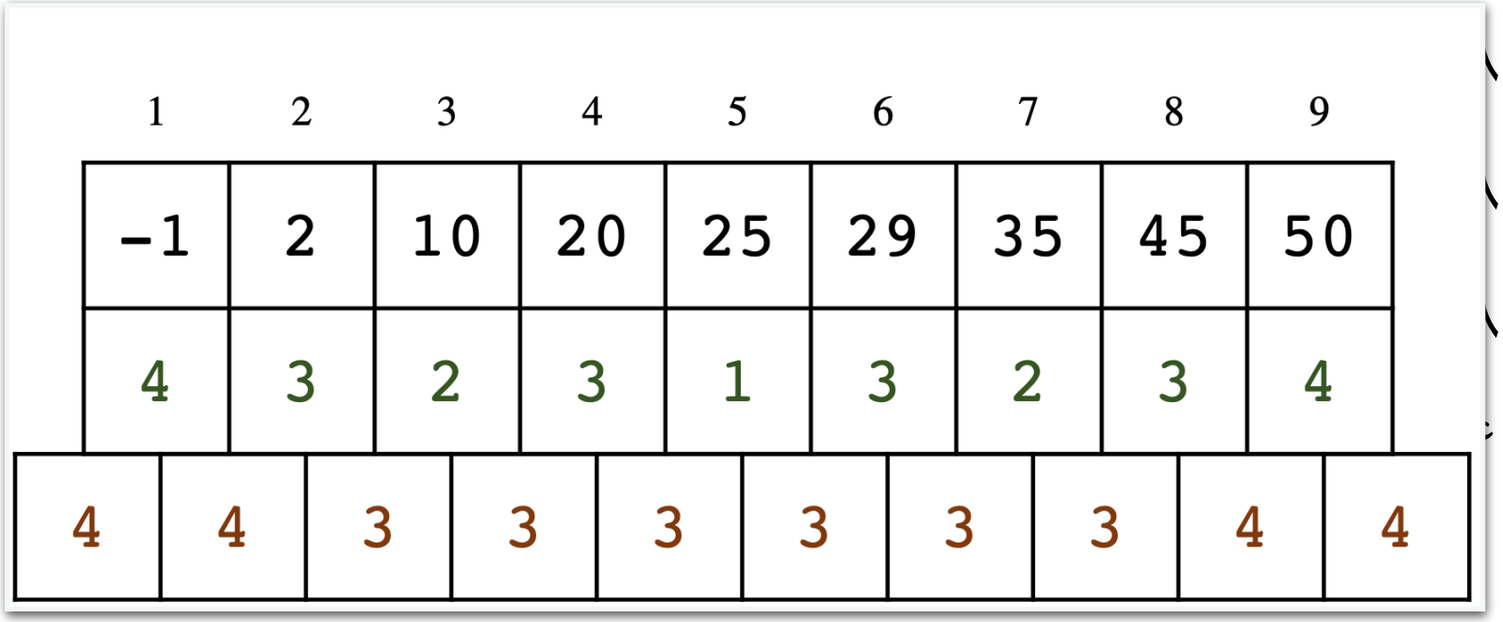
۳) موفق ۲,۸ ناموفق ۳,۴

۴) موفق ۳,۴ ناموفق ۳,۴

سوال ۴.

گزینه ۳.

اگر الگوریتم جستجوی دودویی را برای جستجوی عناصر آرایه $A[1..9] = [-1,2,10,20,25,29,35,45,50]$ به کار ببریم، میانگین تعداد مقایسه‌ها برای جستجوی موفق و ناموفق، به طور تقریبی چقدر است؟



۱) موفق

۲) موفق

۳) موفق

۴) موفق

سوال ۵.

ارشد ۹۸

در عبارت محاسباتی زیر، عملگر + به عملگر × اولویت دارد، این عبارت معادل کدام عبارت پیشوندی زیر است؟

$$((2 + 3) \times 4 + 5 \times (6 + 7) \times 8) + 9$$

۱) $+ \times + + 234 \times \times 5 + 6789$

۲) $+ \times \times \times + 23 + 45 + 6789$

۳) $+ + \times + 234 \times \times 5 + 6789$

۴) $\times \times \times + + 23 + 45 + 6789$

سوال ۵.

گزینه ۲.

در عبارت محاسباتی زیر، عملگر + به عملگر × اولویت دارد، این عبارت معادل کدام عبارت پیشوندی زیر است؟

$$((2 + 3) \times 4 + 5 \times (6 + 7) \times 8) + 9$$

ابتدا بر اساس حق تقدم عملگر ها، پرانتزگذاری میکنیم:

$$(((2 + 3) \times (4 + 5) \times (6 + 7) \times 8) + 9)$$

$$(((((2 + 3) \times (4 + 5)) \times (6 + 7)) \times 8) + 9)$$

سوال ۵.

گزینه ۲.

در عبارت محاسباتی زیر، عملگر + به عملگر × اولویت دارد، این عبارت معادل کدام عبارت پیشوندی زیر است؟

$$(((((2 + 3) \times (4 + 5)) \times (6 + 7)) \times 8) + 9)$$

عملگر را به ابتدای پرانتز و قبل عملوندها منتقل می‌کنیم:

$$(+ ((+2\ 3) \times (+4\ 5) \times (+6\ 7) \times 8) \quad 9)$$

$$(+ (\times (\times (\times (+2\ 3)(+4\ 5))(+6\ 7)8) \quad 9)$$

$$= + \times \times \times +2\ 3 + 4\ 5 + 6\ 7\ 8\ 9$$

سوال ۶.

ارشد ۹۸

اعداد ۱ تا ۶ به ترتیب صعودی در ورودی یک پشته داده شده است. کدام یک از موارد زیر را نمی‌توان با هیچ ترتیبی از درج و حذف در خروجی داشته باشیم؟(اعداد را از چپ به راست بخوانید)

۱) ۱۲۳۵۶۴

۲) ۲۱۵۳۴۶

۳) ۳۲۴۶۵۱

۴) ۴۳۲۱۶۵

سوال ۶.

گزینه ۲.

اعداد ۱ تا ۶ به ترتیب صعودی در ورودی یک پشته داده شده است. کدام یک از موارد زیر را نمی‌توان با هیچ ترتیبی از درج و حذف در خروجی داشته باشیم؟(اعداد را از چپ به راست بخوانید)

۱) ۱۲۳۵۶۴

۲) ۲۱۵۳۴۶ ← قابل ساخت نیست!

۳) ۳۲۴۶۵۱

۴) ۴۳۲۱۶۵

سوال ۷.

ارشد ۹۸

فرض کنید k پشته $S_1, \cdots, S_k$ را در اختیار داریم، تنها اعمال مجاز گرفتن یک ورودی و درج کردن آن داخل پشته S_1؛ حذف یک عنصر از S_k و قرار دادن آن در خروجی، حذف کردن یک عنصر از پشته S_i (برای $i < k$) و درج کردن آن داخل پشته S_{i+1} است. فرض کنید اعداد $1, \cdots, n$ به ترتیب کوچک به بزرگ به عنوان ورودی داده می‌شود؛ کوچکترین k که می‌توان همه‌ی جایگشت‌های $1, \cdots, n$ را با اعمال مجاز گفته‌شده تولید کرد، در بین گزینه‌ها کدام است؟

۱) 1

۲) 2

۳) n

۴) $n - 1$

سوال ۷.

ارشد ۹۸

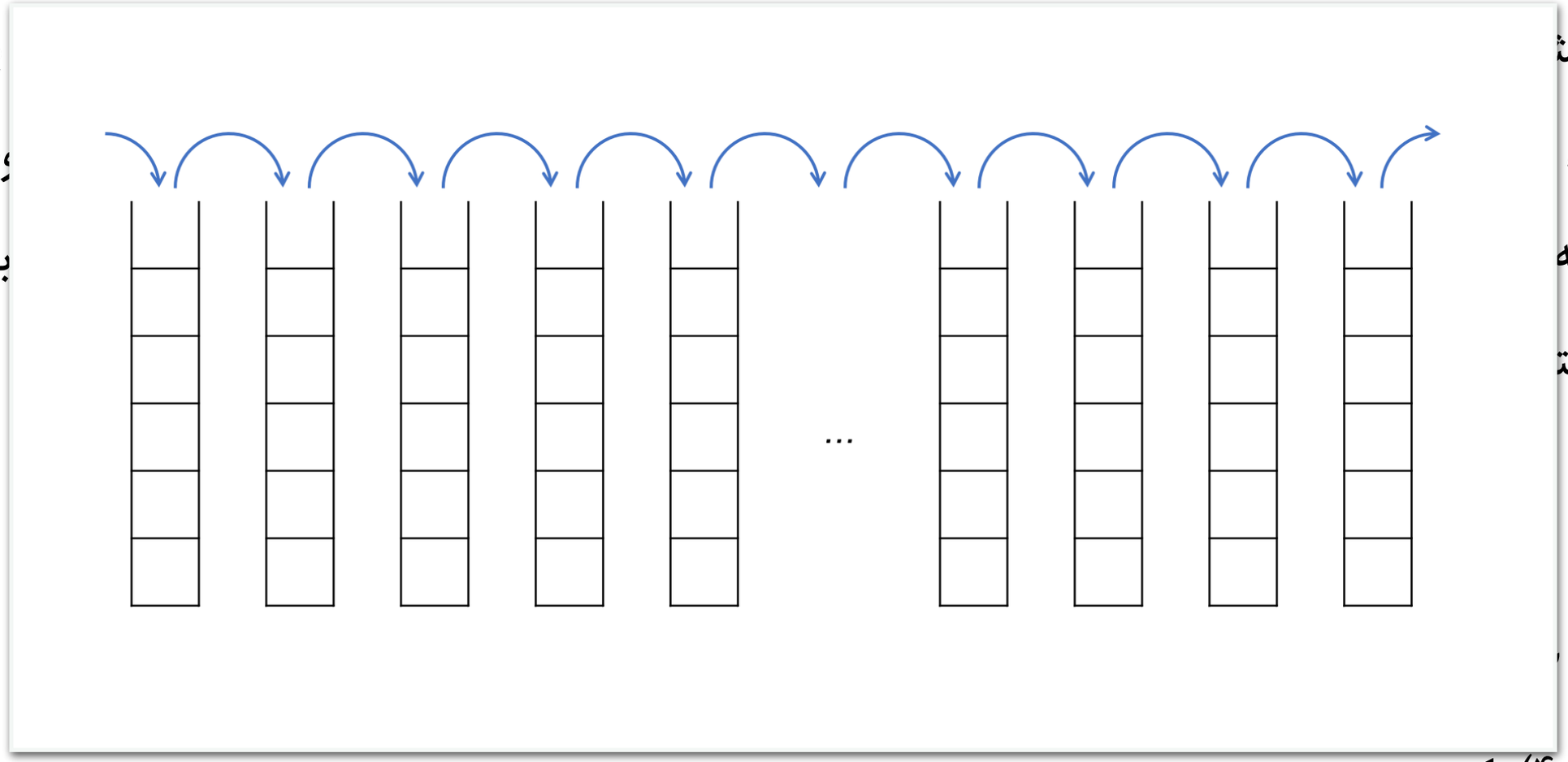
فرض کنید k پشته $S_1, \cdots, S_k$ را در اختیار داریم، تنها اعمال مجاز گرفتن یک ورودی و درج کردن آن

داخل پش پشته S_i

(برای k وچک به

با اعمال بزرگ به

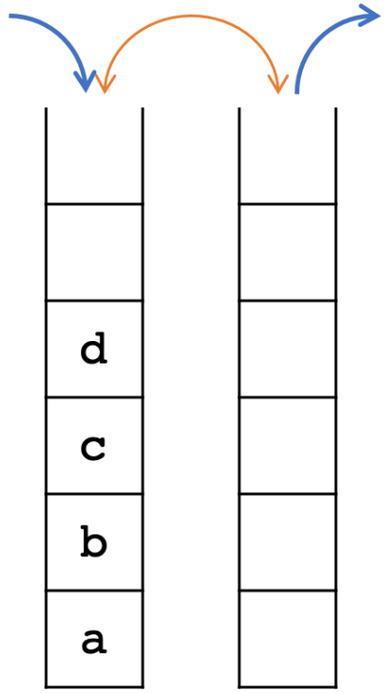
مجاز گفت



$n-1$ (۴

سوال ۷.

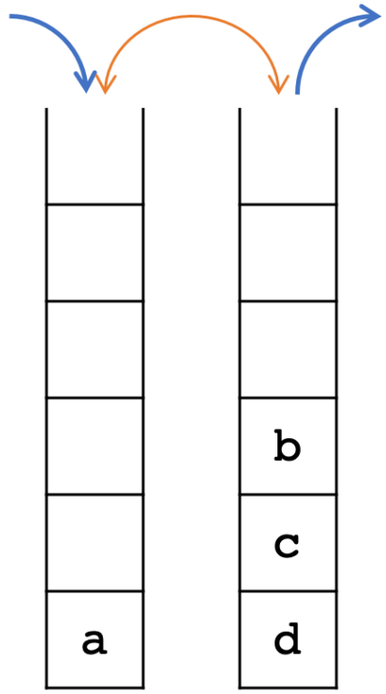
ساختن آرایه با ۲ پشته



سوال ۷.

ساختن آرایه با ۲ پشته

مثلا اگر عنصر دوم رو نیاز داشته باشیم:

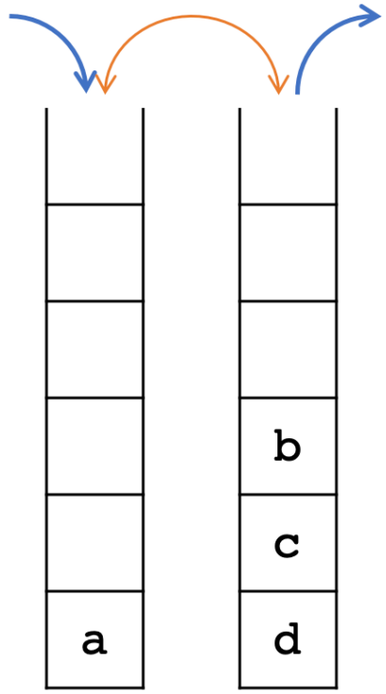


مرتبه زمانی دسترسی به عناصر: داغون! $O(n)$

سوال ۷.

ساختن آرایه با ۲ پشته

مثلا اگر عنصر دوم رو نیاز داشته باشیم:

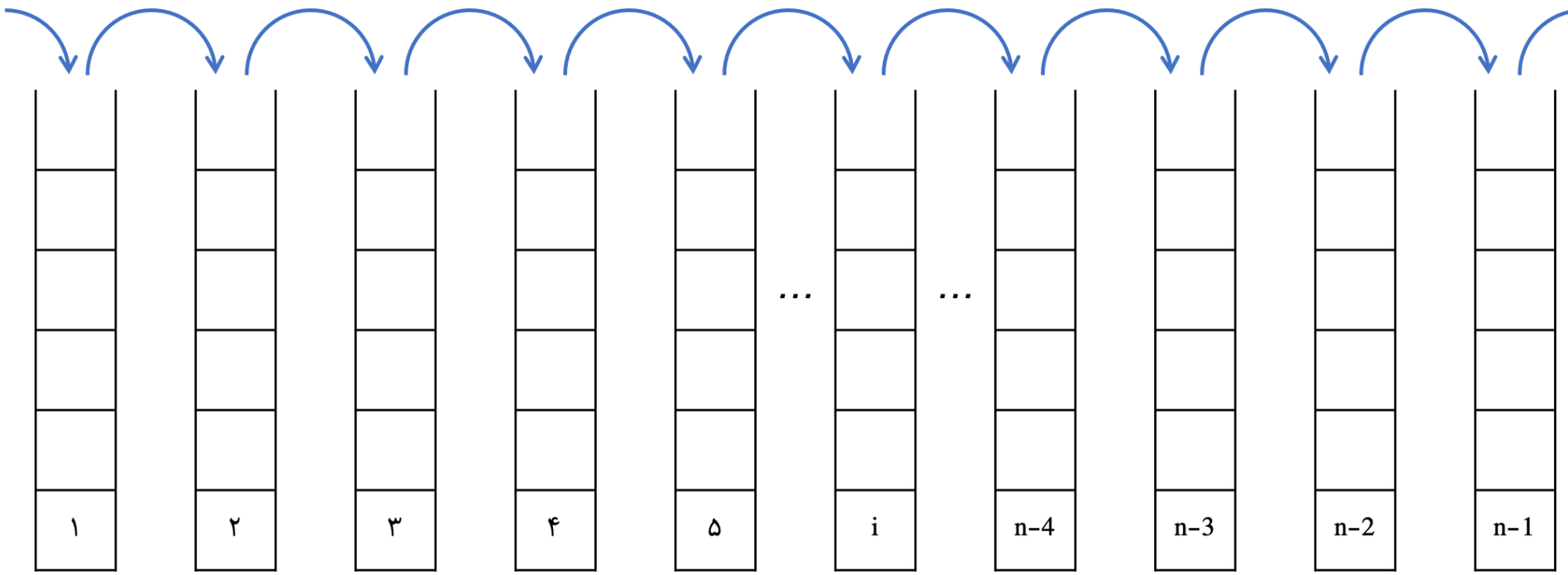


مرتبه زمانی دسترسی به عناصر: داغون! $O(n)$ ← مهمه؟ نه!

سوال ۷.

گزینه ۴.

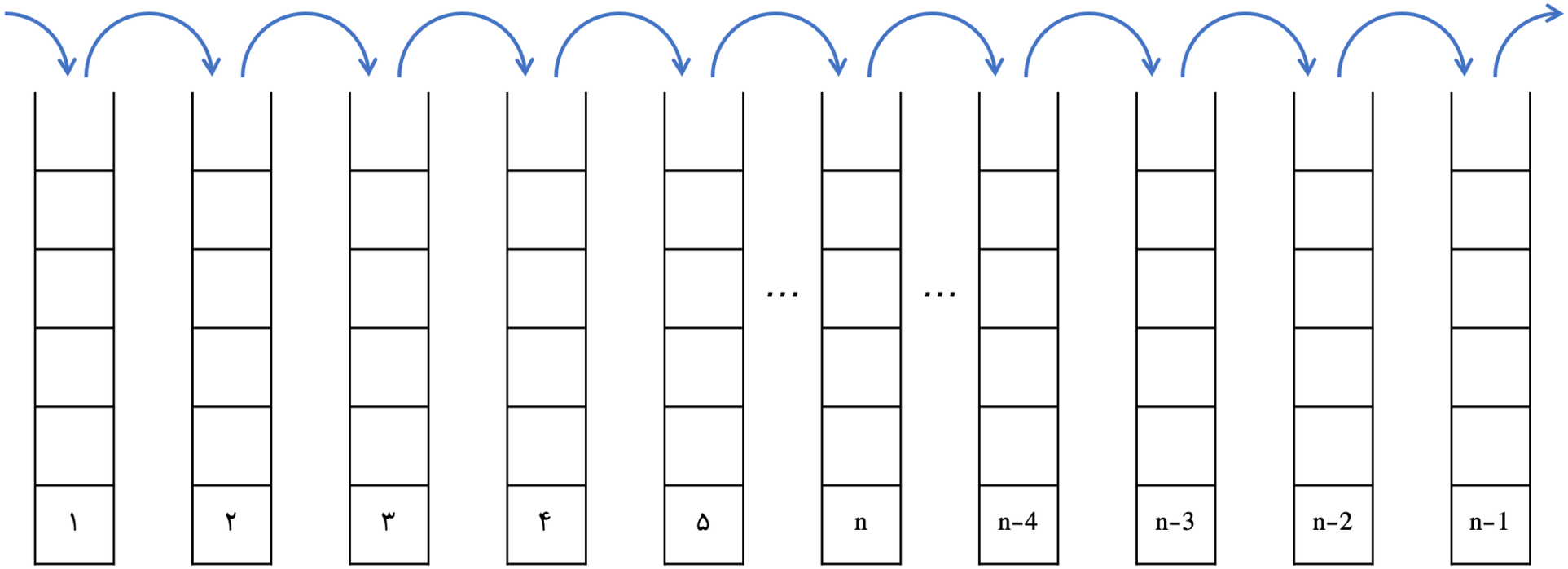
اگر عنصر اول جایگشت خروجی n بود:



سوال ۷.

گزینه ۴.

اگر عنصر اول جایگشت خروجی i بود:



سوال؟

در گوش خواب آلوده جاده فریاد می کشد!

رهگذر