حل تمرین سری سوم(صف و لیست‌پیوندی)

ساختمان داده ها و الگوریتم

سجاد هاشمیان، بهار ۱۴۰۰

سوال ۱.

ارشد ۹۳

برای ساخت یک صف Q از دو پشته S_1 و S_2 استفاده می‌کنیم. برای درج x در انتهای Q ، عمل $Push(S_1, x)$ را انجام می‌دهیم. برای حذف یک عنصر از ابتدای Q ، اگر S_2 خالی نباشد، عمل $Pop(S_2)$ را انجام می‌دهیم. در غیر این صورت، همه‌ی عناصر S_1 را به ترتیب Pop کرده و $Push(S_2)$ می‌کنیم. اکنون عمل Pop بر روی S_2 عنصر ابتدایی Q را بر می‌گرداند.

اگر بر روی Q که در ابتدا خالی است، ۱۰۰ عمل صورت گیرد، حداکثر هزینه چه مقدار خواهد بود؟

۱) 150

۲) 151

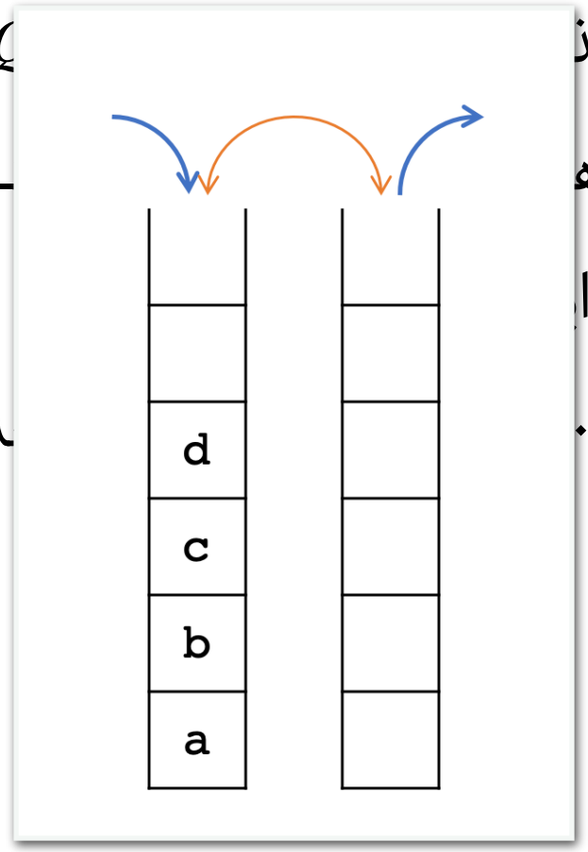
۳) 199

۴) 200

سوال ۱.

ارشد ۹۳

بـرای سـاخـت یک صـف Q از دو پشـته S_1 و S_2 اسـتفاده می‌کنیم. بـرای درج x در انـتهای Q ، عـمل $Push(S_1, x)$ را انجام می‌دهیم. برای حذ Q ، اگر S_2 خالی نباشد، عمل $Pop(S_2)$ را انـجام می‌دهیم. در غیر این صـورت، ه ـرتیب Pop کرده و $Push(S_2)$ می‌کنیم.



اکنون عمل Pop بر روی S_2 عنصر ابتدا

اگر بر روی Q که در ابتدا خالی است، ۰ داکثر هزینه چه مقدار خواهد بود؟

۱) 150

۲) 151

۳) 199

۴) 200

سوال ۱.

گزینه ۳

- هزینه $Push$ در هر شرایطی $O(1)$ است.
- هزینه Pop درصورتی که S_2 خالی نباشد، $O(1)$ است.
- هزینه Pop در صورتی که S_2 خالی باشد، $O(n_{S_1})$ است.
- پس باید سعی کنیم S_2 را تا جای ممکن خالی نگه داریم و سر آخر Pop کنیم:

$$Push, Push, Push, \cdots, Pop$$

$$T(100) = \underbrace{99 \times 1}_{Push} + \underbrace{99}_{\text{انتقال از } S_1 \text{ به } S_2} + \underbrace{1}_{Pop}$$

سوال ۲.

دکتری ۹۵

داده ساختار صف با سه عملیات افزودن به ابتدای صف، حذف از انتهای صف و استخراج عنصر کمینه را در نظر بگیرید.

بهترین پیاده‌سازی ممکن برای این داده ساختار هر یک از سه عملیات فوق را به‌صورت «سرشکن» در چه زمانی پشتیبانی می‌کند؟

۱) هر سه عملیات $O(1)$

۲) هر سه عملیات $O(\log n)$

۳) درج و حذف $O(1)$ و استخراج کمینه $O(\log n)$

۴) درج و حذف $O(\log n)$ و استخراج کمینه $O(n)$

سوال ۲.

دکتری ۹۵

داده ساختار صف با سه عملیات افزودن به ابتدای صف، حذف از انتهای صف و استخراج عنصر کمینه را در نظر بگیرید.

				5	11	2	4	9
POP					5	11	2	4
POP MIN						5	11	4
PUSH 6					6	5	11	4

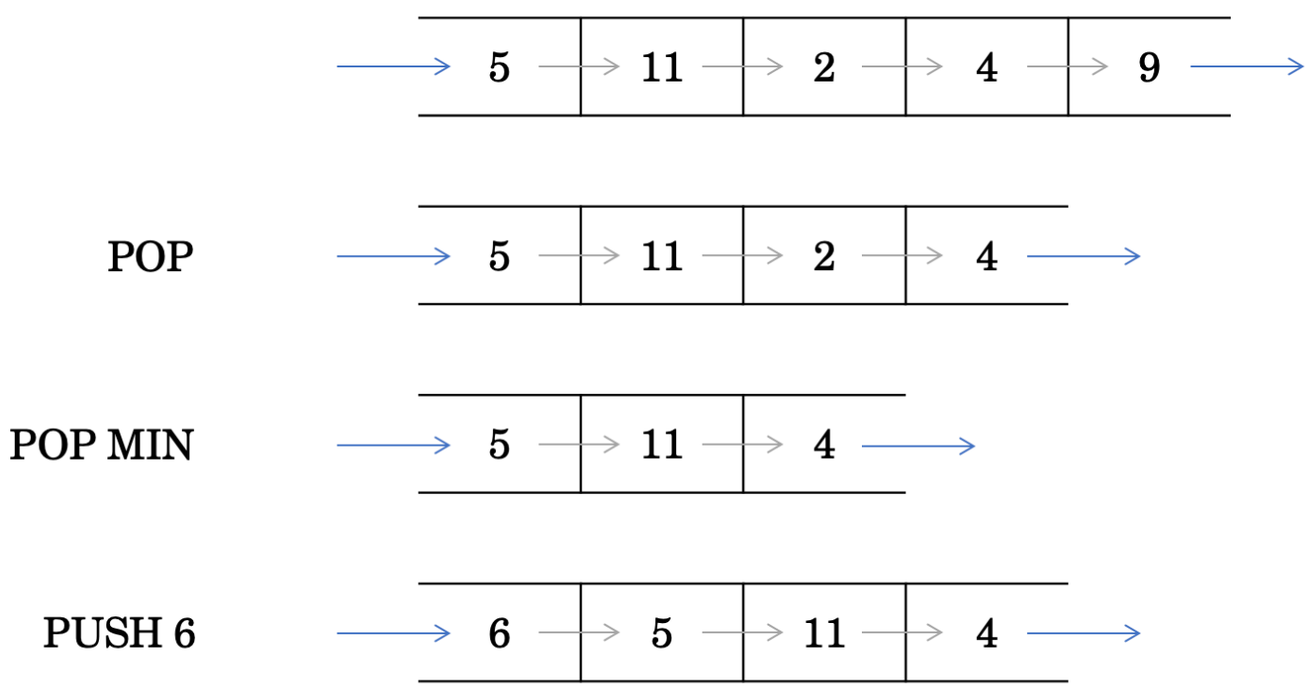
بهترین پی
شکن» در
چه زمانی
۱) هر سه
۲) هر سه
۳) درج و
۴) درج و

سوال ۲.

دکتری ۹۵

فرض کنید که صف ما با یک لیست پیوندی دوطرفه پیاده‌سازی شده باشد.

بنابراین هر خانه از یک دوتایی $(key, *next)$ تشکیل شده است و برای لیست نیز یک اشاره‌گر $begin$ داریم.

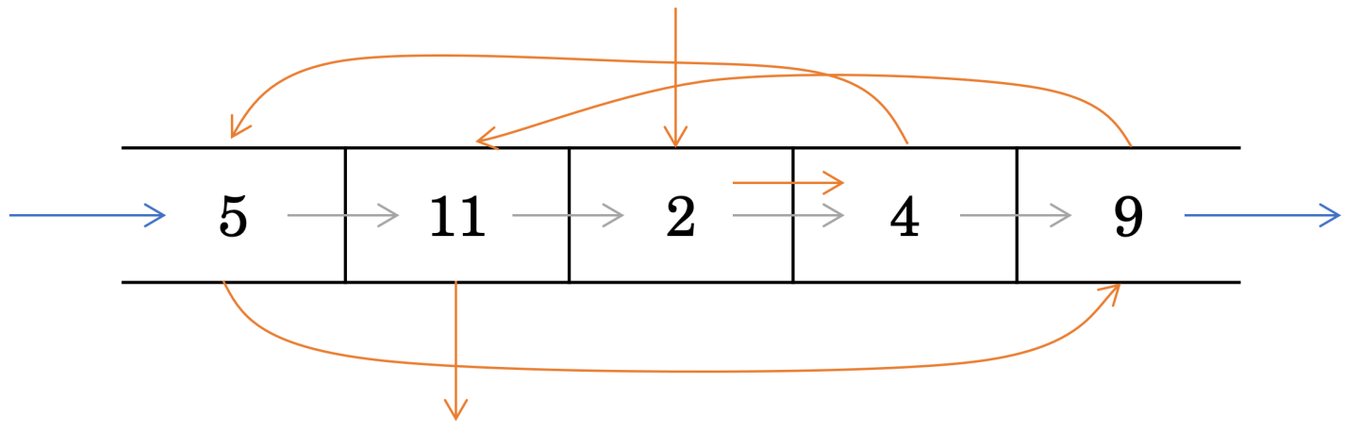


سوال ۲.

گزینه ۱

معماری خانه‌ها را به $(key, *next, *next\ min)$ تغییر می‌دهیم.

حالا به خانه کمینه از $O(1)$ دسترسی داریم و برای حذف و درج گره جدید هم مشکلی نداریم.

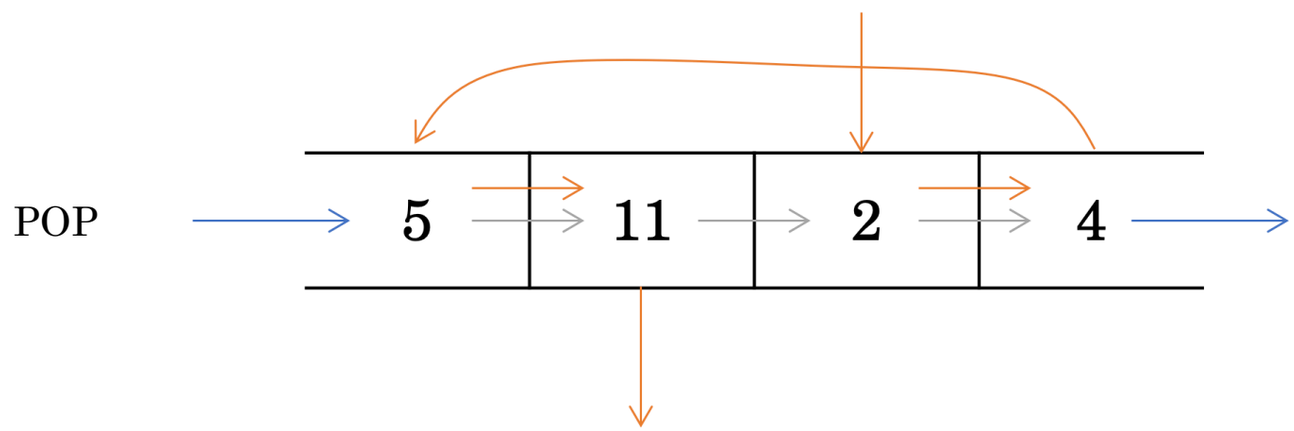


سوال ۲.

گزینه ۱

معماری خانه‌ها را به $(key, *next, *next\ min)$ تغییر می‌دهیم.

حالا به خانه کمینه از $O(1)$ دسترسی داریم و برای حذف و درج گره جدید هم مشکلی نداریم.

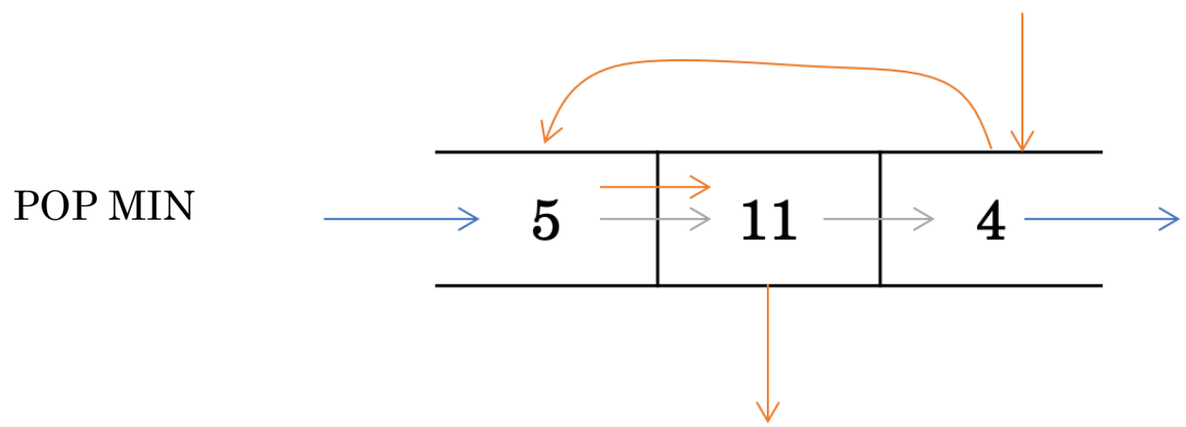


سوال ۲.

گزینه ۱

معماری خانه‌ها را به $(key, *next, *next\ min)$ تغییر می‌دهیم.

حالا به خانه کمینه از $O(1)$ دسترسی داریم و برای حذف و درج گره جدید هم مشکلی نداریم.



سوال ۲.

گزینه ۱

برای عملیات حذف عادی و حذف کمینه که مشکلی نداشتیم.

ادعا می‌کنیم که برای عملیات درج به همان سبک عمومی نیز مشکلی نداریم.

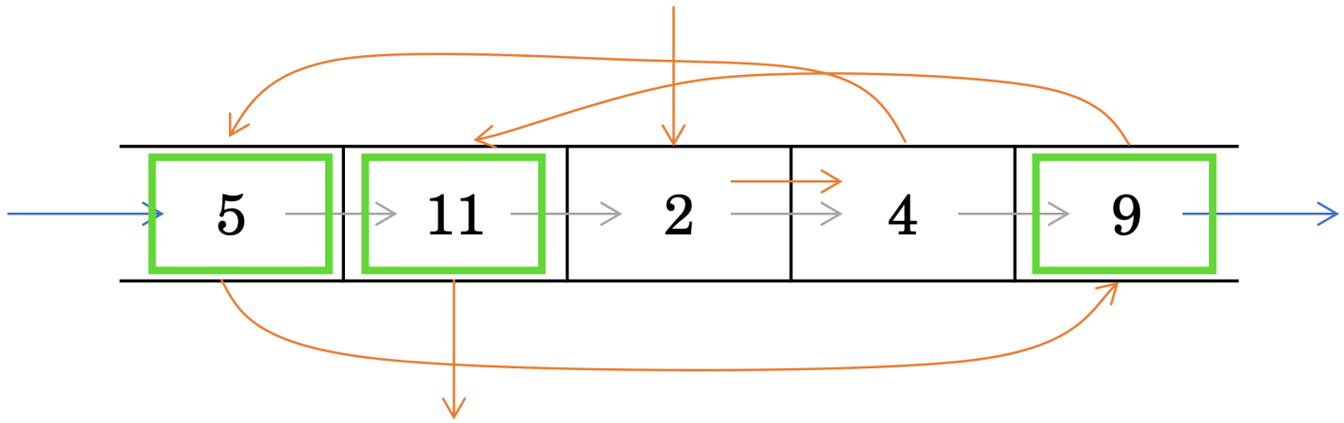
```
def insert_in_minQ(value):
    x=Q.min()
    while(Q[x].nextmin>=value):
        x=(Q[x].nextmin).nextmin
    (Q.nextmin).nextmin=value
    Q.nextmin=value
```

سوال ۲.

گزینه ۱

برای اثبات نیز کافیست همانند تحلیل عملیات درج در **لیست پرشی تصادفی** عمل کنیم.

تابع پتانسیل را «تعداد اشاره‌گر های nextmin در زیر لیست که بعد از x می‌آید» تعریف می‌کنیم:



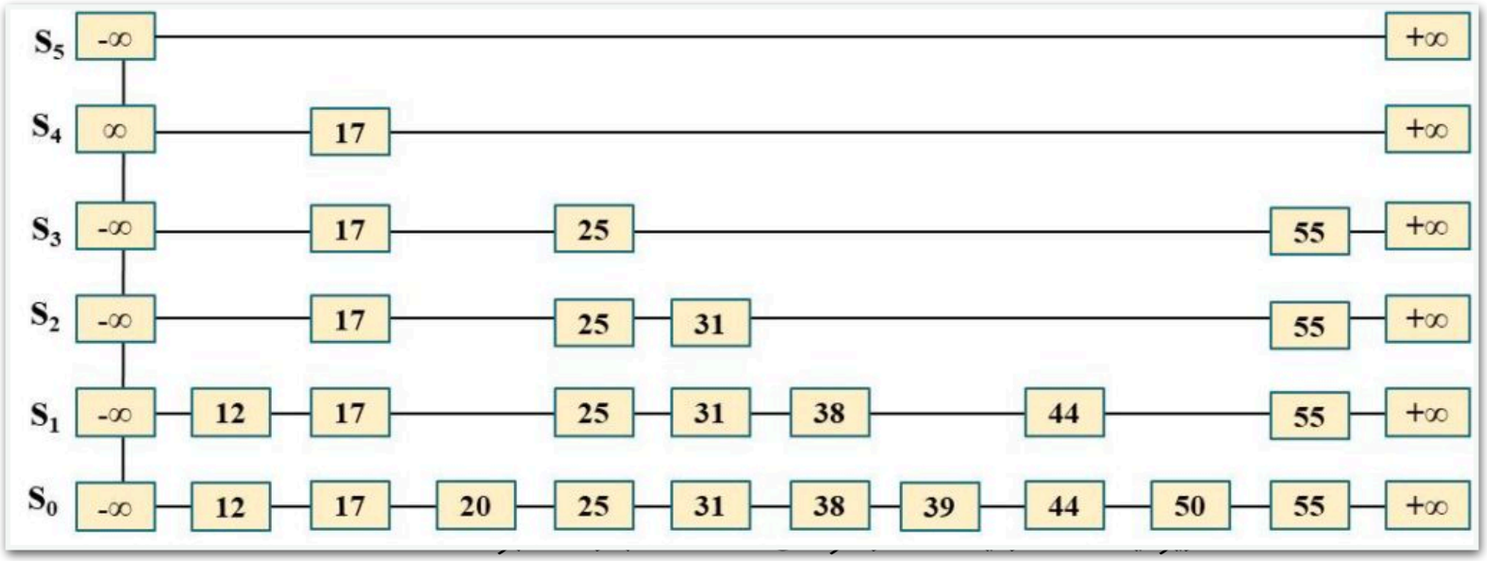
مثلا زیر لیست ۵(در لیست اشاره‌گر های nextmin) با رنگ سبز نشان داده شده است.

سوال ۲.

گزینه ۱

برای اثبات نیز کافیست همانند تحلیل عملیات درج در **لیست پرشی تصادفی** عمل کنیم.

تابع پتانسیل را «تعداد اشاره‌گر های nextmin در زیر لیست که بعد از x می‌آید» تعریف می‌کنیم:



سوال ۳.

ارشد ۹۴

فرض کنید صف Q با یک آرایه‌ی حلقوی به اندازه‌ی m پیاده‌سازی شده است که اندیس‌های آن از صفر تا $m-1$ است و عناصر آن به صورت چرخه‌ای و در جهت ساعتگرد ذخیره شده‌اند. مولفه‌های $front(Q)$ و $rear(Q)$ به ترتیب اندیس اولین عنصر و عنصر بعد از آخرین عنصر صفر را ذخیره می‌کنند. تعداد عناصر داخل صف و شرط پر بودن صف کدام گزینه زیر است؟

۱) $rear(Q) - front(Q) + 1 \mod m$ و $front(Q) = rear(Q)$

۲) $rear(Q) - front(Q) \mod m$ و $front(Q) = rear(Q)$

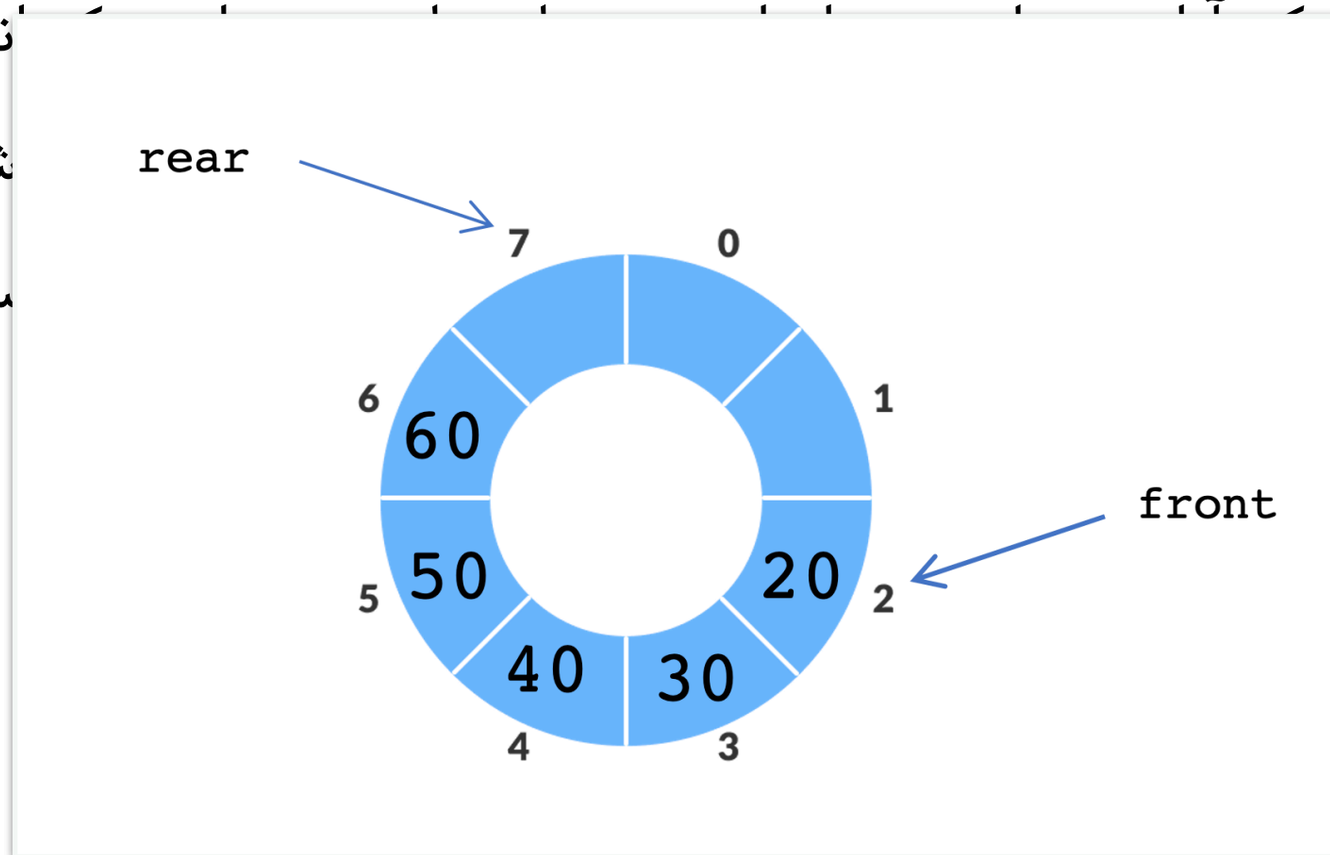
۳) $rear(Q) - front(Q) + 1 \mod m$ و $front(Q) = rear(Q) + 1 \mod m$

۴) $rear(Q) - front(Q) \mod m$ و $front(Q) = rear(Q) + 1 \mod m$

سوال ۳.

گزینه ۴

فرض کنید صف Q با [illegible] اندیس‌های آن از صفر تا $m-1$ است و عنا[illegible] شده‌اند. مولفه‌های [illegible] $front(Q)$ و $rear(Q)$ [illegible] صفر را ذخیره می‌کنند.



تعداد عناصر داخل ص[illegible]

۱) $(Q) + 1 \mod m$

۲) $front(Q) \mod m$

۳) $(Q) + 1 \mod m$

۴) $rear(Q) - front(Q) \mod m$ و $front(Q) = rear(Q) + 1 \mod m$

سوال ۴.

ارشد ۹۴

یک لیست پیوندی خطی با دو اشاره‌گر به ابتدا و انتهای آن در نظر بگیرید. کدام یک از اعمال زیر وابسته به طول لیست می‌باشد؟

۱) حذف عنصر از ابتدا لیست

۲) اضافه‌کردن عنصر به ابتدا لیست

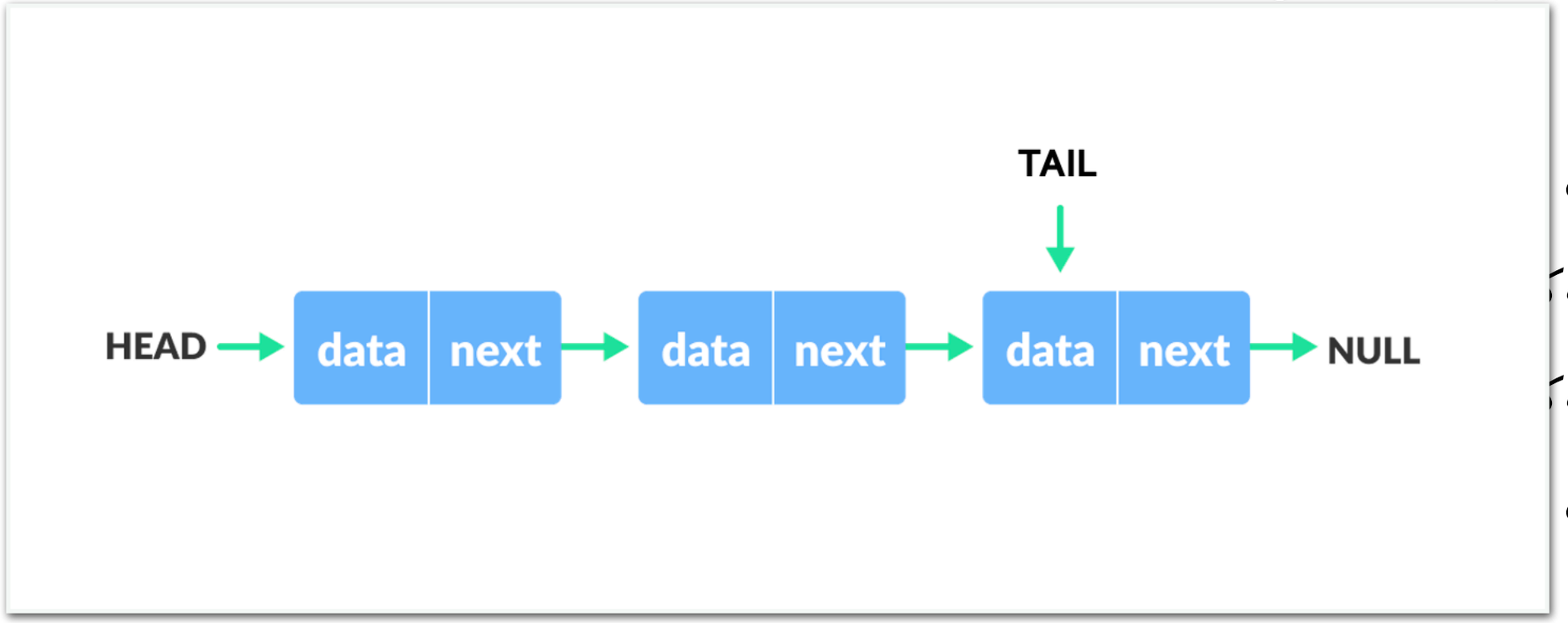
۳) اضافه‌کردن عنصر به انتهای لیست

۴) حذف عنصر از انتهای لیست

سوال ۴.

گزینه ۴

یک لیست پیوندی خطی با دو اشاره‌گر به ابتدا و انتهای آن در نظر بگیرید. کدام یک از اعمال زیر وابسته به طول لیست می‌باشد؟



۱) حذف

۲) اضافه

۳) اضافه

۴) حذف

سوال ۵.

ارشد ۹۳

روی لیست پیوندی و دوسویه‌ی Q که عناصر آن عدد هستند و اشاره‌گر به عنصر اول و آخر آن را داریم، اعمال زیر تعریف شده‌اند:

$Delete(k)$: k عنصر ابتدای Q را به ترتیب حذف می‌کند.

$Append(C)$: عنصر آخر Q را نگاه می‌کند، اگر مقدارش از C بیشتر بود آن را حذف می‌کند. این کار را تکرار می‌کند تا عنصر انتهایی کمتر یا مساوی C شود (یا Q تهی شود). در آن صورت عنصر C را به انتهای صف درج می‌کند.

اگر دنباله‌ای از n تا از این دو عمل را با ترتیب دلخواه روی یک لیست تهی Q انجام دهیم. مجموع کل هزینه‌ها به کدام گزینه زیر نزدیک‌تر است؟

۱) $n-k$

۲) n

۳) $2n$

۴) $3n$

سوال ۵.

ارشد ۹۳

روی لیست پیوندی و دوسویه‌ی Q که عناصر آن عدد هستند و اشاره‌گر به عنصر اول و آخر آن را داریم، اعمال زیر تعریف شده‌اند:

$Delete(k)$: k عنصر ابتدای Q را به ترتیب حذف می‌کند.

$Append(C)$: عنصر آخر Q را نگاه می‌کند، اگر مقدارش از C بیشتر بود آن را حذف می‌کند. این کار را تکرار می‌کند تا عنصر انته

.

اگر دنباله

زینه‌ها به

کدام گزینه

۱) $n-k$

۲) n

۳) $2n$

۴) $3n$

	11	23	62	45	91	83	3	32	74	15
Append(4)	11	23	62	45	91	83	3	4		
Delete(3)				45	91	83	3	4		

سوال ۵.

چشم ها را باید شست، جور دیگر باید دید.

سهراب سپهری

- هزینه عمل $Delete(k)$ وابسته به تعداد عناصر موجود n_t و k هست، که هر دو متغییر اند!
  - $O(min\{k, n_t\})$
- هزینه عمل $Append(C)$ هم وابسته به مقدار C و همچنین تعداد عناصر موجود n_t هست!
  - $O(min\{C, n_t\})$

- خب حالا چجوری مجموع هزینه اجرای n عمل رو بدست بیاریم؟

سوال ۵.

گزینه ۳

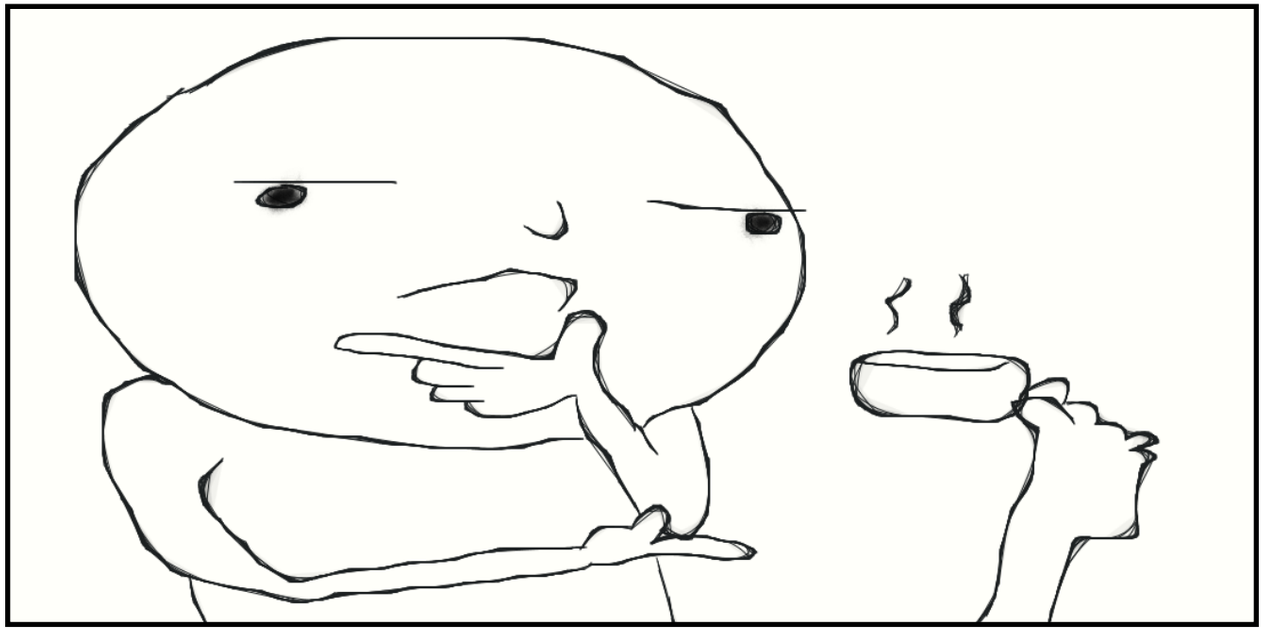
برای حساب کردن مجموع هزینه، بجای محاسبه هزینه هر عمل و کثافت بازی(مثلا میانگین اینا)، میتونیم هزینه رو برای هر عنصر از سیستم‌مون در نظر بگیریم و مجموع رو برای اون‌ها حساب کنیم.

- هر عنصر C توسط $Append$ یک بار اضافه می‌شه.
- هر عنصر C یک‌بار و فقط یک‌بار توسط $Append$ یا $Delete$ حذف می‌شه.

بنابراین برای هر عنصر حداکثر ۲ عمل و حداقل ۱ عمل انجام میشه.

این یعنی کلا به‌طور متوسط $2n$ عملیات انجام می‌دیم.

سوال؟

ای بی‌حافظه‌شده پس از نوبت‌ها شوک برقی!

رضا براهنی