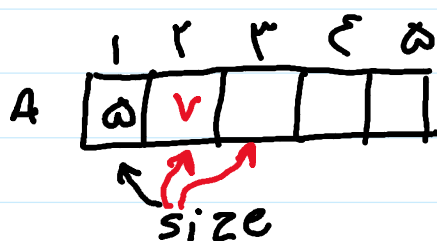


موضوع: داده ساختارهای ابتدایی: آرایه - لیست پیوندی - اشک و صف - لیست

مسئله ۱) درودی: تعدادی دستوره صورت $add(x)$ و ما باید این اعداد را به صورت مرتبانه داریم.

* راه حل: یک آرایه A نگه داریم و اعضا را به آن اضافه کنیم:



$add(5)$

$add(7)$

* مشکل این راه؟ اندازه آرایه.

روش دوم، آرایه پویا به دو برابر کردن

ابتدا با یک آرایه با اندازه ۱ شروع کنیم. در زیر $add(x)$

* برای اضافه کردن x جاهست: x را اضافه می کنیم

نست:

۱۱ *

- ۱- یک آرایه جدید با اندازه ۲ برابر ایجاد می کنیم
- ۲- تمام عناصر آرایه قبلی را به آن انتقال می دهیم
- ۳- x را در آرایه جدید درج می کنیم.

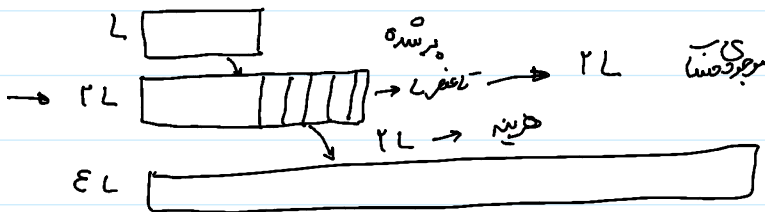
ادعا: هزینه سرسختی هر بار عملیات اضافه کردن: $O(1)$ است

ادعا: هزینه سرسری هر بار عمليا اضافه کردن: $O(1)$ است
 \downarrow
 $\text{add}(i)$

روش حساداری: به ازای اولین بار که عنصر x در آرایه اضافه می شود: ۳ ریال
 محاسبات حساداری
 ۲ ریال

برای پرداخت هزینه انتقال x بین آرایه ها: ۱ ریال از حساداری وقت کن

* حساداری به خالی نمی شود؟



۳ روش تابع پتانسیل: $\phi_i = 2i - L$ طول آرایه: L

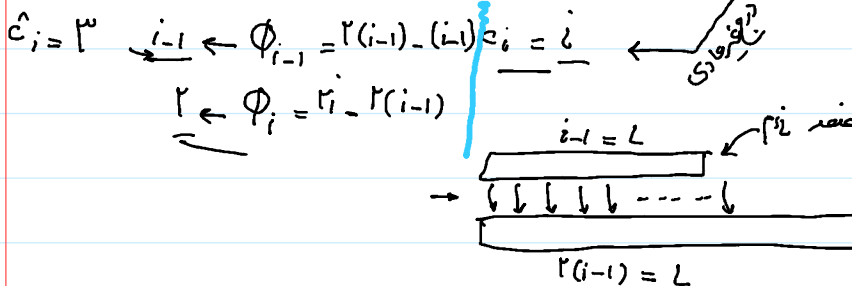
$\phi_0 = 0$ $\phi_i \geq 1$ همیشه نصف + ۱ از آرایه پر است

هزینه عمل i ام $\hat{c}_i = c_i + \phi_i - \phi_{i-1}$

$$\sum_i \hat{c}_i = \sum_i c_i + \frac{\phi_n}{1} - \frac{\phi_0}{0}$$

$$\sum_i \hat{c}_i \geq \sum_i c_i$$

$\hat{c}_i = 3 \Leftarrow \phi_i - \phi_{i-1} = 2 \quad c_i = 1$



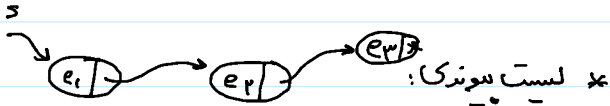
$$\sum_i \hat{c}_i = 3n \geq \sum_i c_i$$

داده ساختارهای ساده. صرف و اسفند:

۱. داده ساختار پایه ای: آرایه - لیست پیوندی

* آرایه: دنباله یست سرهم از خانه ها حافظه

* به عنصر از آرایه در $O(1)$ دسترسی داریم
 * طول آرایه ثابت است.



* لیست پیوندی: $R(n) \leftarrow R(k)$: دسترسی به عنصر k از n
 * طول لیست نامحدود

abstract data type: ADT به صورت مجرد در مورد داده ساختار عملیات

که نیاز دارد توضیح می دهد.

* چگونه ADT را پیاده کنیم.



* استک: stack (LIFO)

عملیات: $Push(x)$: x را وارد استک می کند

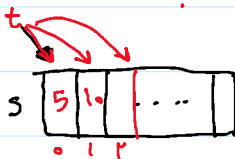
$Pop()$: آخرین عنصر وارد شده را از استک خارج و برمی گرداند

$Top()$

$Size()$

$IsEmpty()$

پیاده سازی استک؟



۱- با استفاده از آرایه:

$Push(5)$ $Push(10)$

t : اندکس عناصر داخل استک.

$O(1)$: $Push(x)$

$O(1)$: $Pop()$

$Push(x_1)$

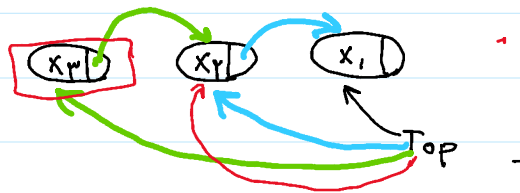
$Push(x_2)$

$Pop()$



Push(x)

Pop()



$O(1)$: Push(x)

$O(1)$: Pop()

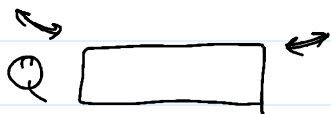
۳. استفاده از آرایه پویا :

* صف Queue ، FIFO .

عملیات : enqueue(obj) : obj را به انتهای صف اضافه می کند.

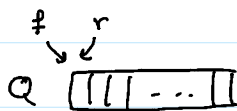
dequeue() : عنصری که در ابتدای صف است را حذف و

به خروجی می دهد.



: size()

: isEmpty()



۱. پیاده سازی با آرایه :

f: front

r: rear



enq(x)

enq(10)

deq

enq(x) {

Q[r] = x

(r++) % N

}

deq() {

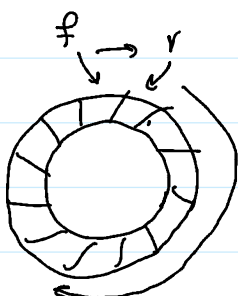
(f++) % N

return

(Q[f])

}

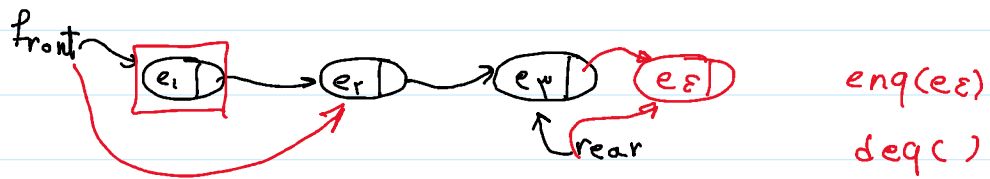
نکته : طول آرایه - ۱



۲. استفاده از صف حلقوی :

۳. پیاده سازی با استفاده از لیست پیوندی

۳- پیاده سازی با استفاده از لیست پیوندی



۴- پیاده سازی با استفاده از آرایه