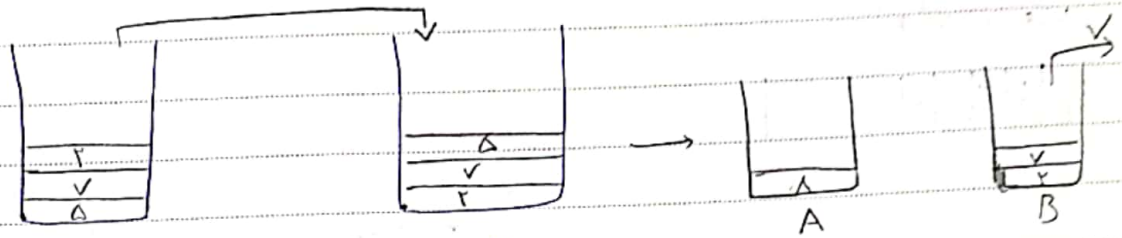
Subject: جلسه ۹ ام
Year. ۱۴۰۰ Month. ۷ Date. ۲۷ ( )

موضوع: درخت ها

* یادآوری: پیاده سازی صف با استفاده از ۲ تا استک



enq(۵)
enq(۷)
enq(۲)
deq() → استک B خالی → ۵ به فروش داده می شود
enq(۸)
deq() → ۷ POP می خورد

enq(a): وارد استک A

deq(): اگر استک B خالی نبود: POP(B)
اگر استک B خالی بود: تمام عناصر را از A POP و به B Push می کند.
سپس یک عنصر از B، POP می کند.

هزینه سرشکن هر deq یا enq برابر با O(1) است.

روش حسابداری: به ازای هر enq، ۴ ریال هزینه می شود.
۱ ریال هزینه Push و ۳ ریال را در حساب میذاریم

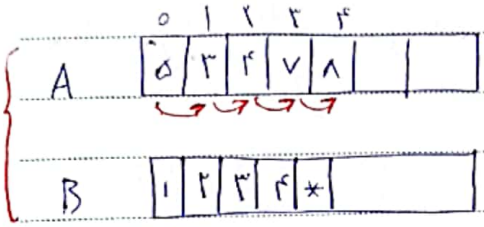
deq → تمام هزینه را از حساب پرداخت می کنیم →
- POP از استک A
- Push در استک B
- یک POP از استک B

همیشه در حساب چیزی باقی وجود دارد، چرا؟ چون وقتی می خواهیم عنصری را در B، Push کنیم از قبل که در A Push شده است هزینه اش پرداخت شده (در حساب ذخیره شده) [هر عنصر حداکثر ۳ حرکت روش انجام میشه]

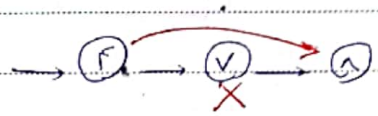
Scanned with CamScanner

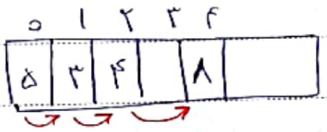
Subject: جلسه ۱۹
Year. ۱۴۰۰ Month. ۷ Date. ۲۷ ( )

* پیاده سازی لیست پیوندی با استفاده از آرایه:

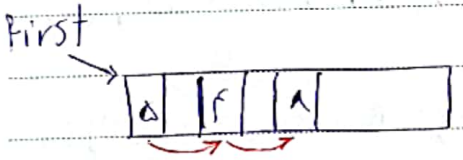


۷ را حذف کنیم:

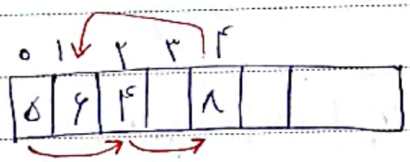




۳ را حذف کنیم:

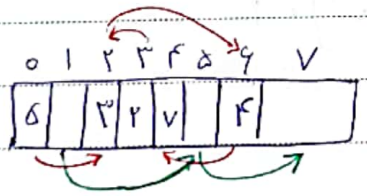


۶ را به انتهای دنباله اضافه کنید:



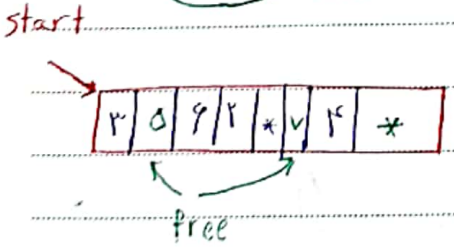
یک خانه‌ی خالی پیدا کنیم.

سوال: چطور یک خانه خالی پیدا کنیم؟ ← O(1)
* یک استک از خانه‌های خالی.
یا (صف)



* بدون حافظه اضافه:

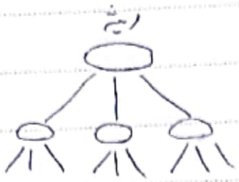
اشاره‌گر اندیس



PAPCO
Scanned with CamScanner

Subject: جلسه ۹ ام
Year. ۱۴۰۰ Month. ۷ Date. ۲۷

درخت ها



برای نشان دادن ساختارهای سلسله مراتبی

Terminology :

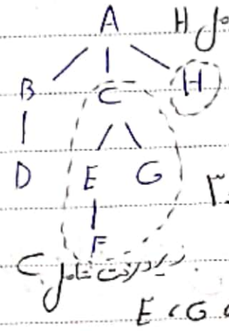
ریشه : راسی از درخت که پدر ندارد

فرزند : راس هایی که پایین تر از یک راس قرار دارند و به طور مستقیم با یک یال بهش متصل می شود.

پدر : راس هایی که با یک یال مستقیم به فرزند وصل می شود.

اجداد : پدر، پدر پدر، پدر پدر پدر و...

برگ : راس بدون فرزند



زیردرخت : خود اون راس و تمام راس هایی که پایین تر از آن وجود دارد

گره داخلی : گره ای که برگ نیست : A_B_C_E

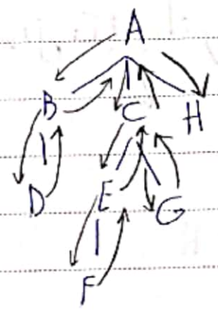
عمق یک راس : تعداد اجداد یک راس راس A : ۰ راس H : ۱ راس F : ۳

ارتفاع درخت : بیشینه عمق + ۱ ← ۴

اولاد : فرزندهای فرزند فرزندها، فرزند فرزند فرزندها و... اولاد C : F، G، E

برادر : دو راس با پدر یکسان B و C

* پیمایش درخت :



ترتیبی برای ملاقات راس های درخت

- پیمایش پیش ترتیب (Preorder)
- پیمایش پس ترتیب (Postorder)

Preorder A : A B D E F G H

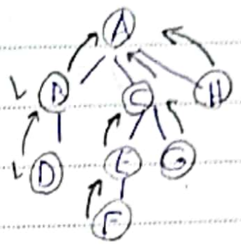
```
Preorder(x) {
   visit(x)
   for all child y of x      ← ترتیب از چپ به راست
     Preorder(y)
}
```

PAPCO

Scanned with CamScanner

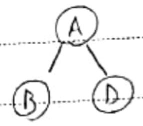
Subject: جلسه ۹ ام
Year. ۱۴۰۰ Month. ۷ Date. ۲۷

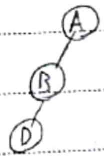
```
Postorder(x) {
  forall child of x
      Postorder(y)
  visit x
}
```



Postorder(A): D B F E G C H A

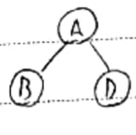
سوال: آیا با داشتن پیمایش Preorder می‌توان درخت را بازسازی کرد؟

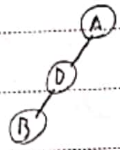




A B D

سوال: آیا با داشتن پیمایش Postorder می‌توان درخت را بازسازی کرد؟



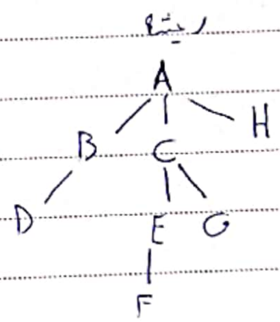


B D A

سوال: آیا با داشتن پیمایش Postorder و Preorder می‌توان درخت را بازسازی کرد؟

Preorder: A B D C E F G H

Postorder: D B F E G C H A



خوب پری اردر هر یه درخت رو بدیمش و با پست اردر قبلش اشتراک بگیریم میشه زیر درخت اول راس

PAPCO

Scanned with CamScanner

Subject: جلسه ۹
Year. ۱۴۰۰ Month. ۷ Date. ۲۷ ( )

{ insert(x)
delet(x)

root() : ADT
Parent(x)
children(x) ← مجموعه فرزندان x
size()

پیاده سازی درخت:

P4PCO

Scanned with CamScanner