

ساختار داده‌ها: داده‌هایی هستند که به شکل خاصی سازماندهی شوند و در یک منطقه یا حافظه سازماندهی داده‌ها را به صورت خاص را ساختار داده می‌گویند به عبارتی دید جهت دریافت داده‌ها را خام توسط کامپیوتر و یا داده ساز ایجاد آلگوریتم‌ها را مختلف و در آن صورت استاده قرار می‌گیرند ساختار داده می‌گویند

آلگوریتم یا آرایه: لیست عناصر داده‌ها را می‌توانیم در یک لیست یا آرایه توسط اندیس ایندکس مشخص می‌شود از آرایه یونی منظور استفاده می‌شود که حافظه کمتری برای نگهداری داده‌ها را میسر می‌آورد

آرایه یک‌بعدی: مجموعه متناهی از زوج‌ها به صورت اندیس مقدار است برین یونی که برای اندیس یک مقدار مربوط به آن وجود دارد

دو بعدی: مجموعه‌ای از n در n عنصر داده‌ها است که هر عنصر آن باید زوج عدد صحیح مانند k آن مشخص می‌شود که این اعداد را به صورت k و n یا $A[k, n]$ نشان می‌دهد آرایه دو بعدی را می‌توان به جدولی تشبیه کرد که در آن n سطرها و n ستون‌ها باشد هر سطر شامل عناصر است که عنصرهای اول آن با هم برابر است و هر ستون شامل عناصر است اندیس‌های دوم آنها برابر هستند آرایه دو بعدی می‌توان به صورت ماتریس نگاشته شود

آرایه: $A: \text{Array}[L, \dots, U]$ of type T به اندازه n یکس فضا اشغال کند و شروع آرایه در این حافظه α آدرس خانه $A[i]$ آدرس خانه و مقدار عناصر آرایه بدون زیر محاسبه می‌شود

$$A[i] = (i - L) * n + \alpha$$

آدرس شروع مقدار حافظه

آرایه یک‌بعدی $(U - L + 1)$ مقدار عناصر

روشن سفیدی $A[i, j] = [(i - L_p)(u_p - L_p + 1) + (j - L_p)] * n + \alpha$
در حالت کلی در آرایه دوبعدی A آدرس خانه $A[i, j]$ بدین صورت بدست می آید

روشن ستونی $A[i, j] = [(j - L_p)(u_p - L_p + 1) + (i - L_p)] * n + \alpha$

α آلفا آدرس شروع آرایه است در صورت عدم بیان آدرس شروع آلفا برابر صفهاست
n برابر صفها نوع عناصر آرایه است در صورت عدم بیان صفها، نوع آرایه
n برابر تعداد می باشد

آرایه دوبعدی زیر آدرس ۳۰۰ به بوی حافظه قرار دارد آدرس عنصر لیست دو و هشت
بدون روشن سفیدی کدام است
روشن سفیدی $List[2, 8] = ?$

List: Array $[-3, \dots, 4, 5, \dots, 20]$ of real $n=7$ $\alpha=300$

اول ابتدای نام در صم

$$List[2, 8] = \left[\left[\underbrace{(2 - (-3))}_{(5)} \underbrace{(2 - 5 + 1)}_{(12)} + \underbrace{(8 - 5)}_{(3)} \right] * 7 \right] + 300$$

$$83 \times 7 = 581 + 300 = 881$$

مثال: آرایه دوبعدی $n[3, \dots, 20, 5, \dots, 8]$ در آدرس ۳۰۰ به بوی حافظه
قرار دارد و هر عنصر آرایه احتیاج به ۷ بایت صفها دارد آدرس عنصر $n[4, 10]$
بدون روشن ستونی کدام است
 $List[4, 10]$

$\alpha = 300$
List: Array $[-5, \dots, 5, 6, \dots, 20]$ of real $n=4$

$$List[4, 10] = \left[\left[\underbrace{(10 - (-3))}_{(13)} \underbrace{(5 - (-5) + 1)}_{(11)} + \underbrace{(4 - (-5))}_{(9)} \right] * 4 \right] + 300$$

$$27 \times 4 = 108 + 300 = 408$$



(sparse)

خفیه جابجایی

ماتریسی که عناصر صفر در آن زیاد باشد ماتریس اسپارسی یا خفیه نامیده می شود

		۱	۲	۳	۴	۵
۱	۱۵	۰	۰	۲۲	۰	-۱۵
۲	۰	۱۱	۳	۰	۰	۰
۳	۰	۰	۰	-۶	۰	۰
۴	۰	۰	۰	۰	۰	۰
۵	۰	۰	۰	۰	۰	۰
۶	۰	۰	۰	۰	۰	۰
۷	۰	۰	۰	۰	۰	۰
۸	۰	۰	۰	۰	۰	۰
۹	۰	۰	۰	۰	۰	۰
۱۰	۰	۰	۰	۰	۰	۰

برای نمایش ماتریس اسپارسی می توان از یک ماتریس کوچکتر ۳ ستون استفاده کرد. در ستون اول شماره سطر و شماره دوم ستون و عناصر غیر صفر است. به این ترتیب در مصفوفه حافظه صرفه جویی می شود. عبارت (I, J, value) به وسیله نمایش یابی و مقدارش مشخص می شود

(I, J, value)

تعداد ستون

تعداد اعداد در ستون	تعداد اعداد در سطر	تعداد اعداد در سطر
۱۵	۰	۰
۲۲	۰	۳
-۱۵	۰	۵
۱۱	۱	۱
۳	۱	۲
-۶	۲	۳
۹۱	۴	۰
۲۸	۵	۲

می تواند از صف شروع شود در جدول بالا شما هم می توانید از ردیف شروع می شود

در این مثال فرض کردیم شماره سطر و ستون از صف شروع می شود همچنین سطر ۱ ۶ ۸ نشان می دهد ماتریس اسپارسی شامل ۱ سطر و ۱ ستون و هشت عنصر غیر صفر است. سطر اول اینطور نوشته می شود و با بقیه سطر ها شماره سطر و شماره ستون مقدار موجود قرار می گیرد



IRAN

> stack < پشته

لیست مرتبی است که عملیات اضافه و حذف کردن از یک طرف آن صورت می گیرد.
 پشته مخفی LIFO ← LAST آخرین عنصر وارد شده اولین عنصر است که خارج می شود به پشته FILO نیز گفته می شود.

LIFO: Last In First Out

FILO: First In Last Out

push به دارد
 pop می دارد



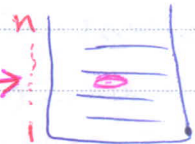
یکه از کاربرد های پشته ذخیره آدرس بازگشت و مسافت مقصید ها محلی در مدار زدن توابع است.
 شماره تدریس راه های پشته که آرایه به پشته به تحول n می باشد.

stack : Array [1 ... n] of item

TOP : 0 ... n

if top = 0 → stack خالی

if top = n → stack پر



عمل اصلی بر روی پشته ها به push و pop می شناسیم n داده ها را در بالای پشته قرار می دهد و عمل pop نیز عنصر بالای پشته در مقصید n ذخیره می کند.
 در اجزای زیر برنامه push بررسی می کنیم که آیا فضای خالی در پشته برای عنصر جدید وجود دارد یا خیر. فضای خالی وجود نداشته باشد با خطا Overflow خطا مواجه می شود. در اجزای زیر برنامه نیز پشته ابتدا باید بررسی کنیم که آیا عنصری در پشته وجود دارد که حذف شود در صورت عدم وجود با خطا Under Flow مواجه می شود.

$n=5$ $A=1$ $B=2$ $C=0$

مثال مقدار نهایی A و B و C چقدر است

push (B)

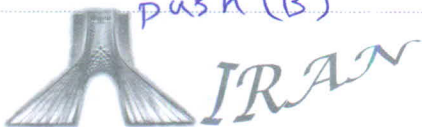
push (A+B)

pop (C)

push (A-B)

push (C)

push (B)



pop (A)

pop (B)

push (AxB)

push (C)

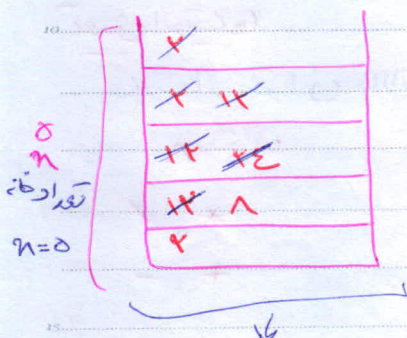
push (A)

pop (B)

pop (C)

pop (A)

جواب



A	B	C
10	7	12
7	12	12
12	2	12
2	2	12
2	2	12

جوابهای



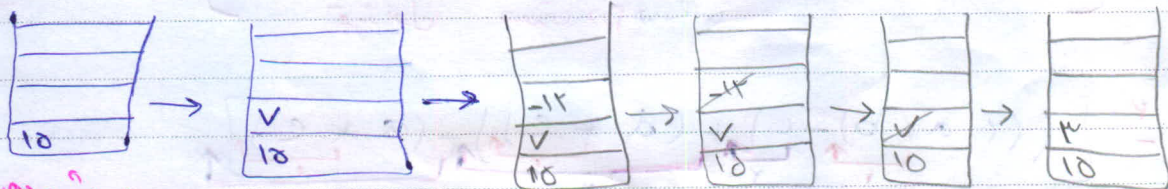
به ترتیب و به روش آریتمی عمل می‌کنیم و می‌بینیم که برقی داریم

تقدیر خاتمه

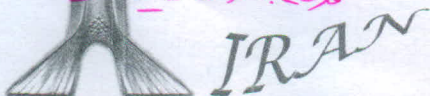
عملیات زیر را از چپ به راست به شکل نمایشی دهید

n=0

push(10), push(7), push(-12), pop(A), pop(B), push(2)



مراحل به مرحله غنی (همه)



عبارت pre Fin - in Fin - post Fin

روش استاندارد برای نوشتن عبارت ریاضی استفاده می‌کنیم in Fin یا میان وشی
 می‌باشد در این روش عملگرها را در بین دو عملوند می‌نویسیم ولی کامپیوترها معمولاً
 بدون پرانتز post Fin یا پسوندی استفاده می‌کنند در این روش هر عملگر بعد از عملوند
 مربوطه ظاهر می‌شود همچنین pre Fin یا پیشوندی قبل از عملگرها ظاهر می‌شوند

بعد از ارفی لید $\rightarrow$ عملگر
 post Fin پسوندی
 وسط قرار میدهد میان وشی
 in Fin میان وشی
 قبل قرار می‌گیرد پیشوندی
 pre Fin پیشوندی

تعیین اولویت‌ها

عبارت از توان جالانین اولویت تقسیم و ضرب مقدم بر جمع و تفریق و جمع یا تفریق بر مقدم

اولی ۸ توان

دومی *

سومی + -

روش دستی تبدیل این in Fin به post Fin به صورت زیر است

- ۱- ابتدا عبارت به طور کامل پرانتز گذاری می‌کنیم (با توجه به اولویت عملگرها)
- ۲- هر عملگر را سمت راست پرانتز بعد از خودش منتقل می‌کنیم
- ۳- تمام پرانتزها را حذف می‌کنیم

پسوندی $\rightarrow$ in Fin

سوال $a / b - c + d * e - a * e$ تبدیل

$$1 - [((a / b) - c) + (d * e) - (a * e)]$$

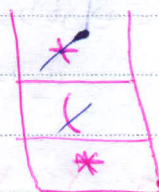
جایگزینی یا اولویت‌ها

$$2 - [((a / b) - c) + (d * e) - (a * e)]$$



$$y \mid a * (b + c) \quad y$$

PostFix: $a \ b \ c \ + \ *$



هر دو داخل آن پراکنده جامد می آید پس Post Fix و پراکنده حذف می شود

Post Fix: $ab + ab / cd * + * ab * -$

مقدم - *
 رقم * /
 اول ٨ ↑
 از جمله می جدول خارج می شود

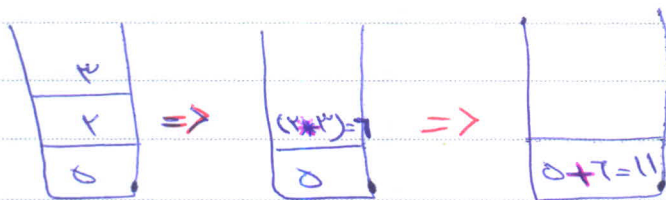
Post Fix ارزیابی شود

الگوریتم صحابه یا ارزش جایی به صورت

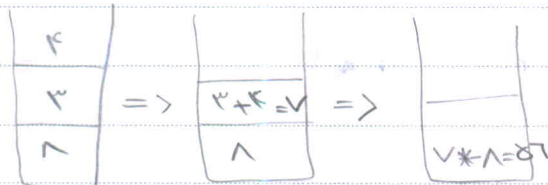
- ۱- عبارت را به صورت حرف به حرف مورد بررسی قرار می دهیم
- ۲- اگر حرف مورد نظر یک عملوند بود آن را داخل پشته قرار می دهیم
- ۳- اگر حرف مورد نظر یک عملند بود به صورت لازم معمولاً دو عملوند از داخل پشته خارج کرده و عمل مورد نظر انجام می دهیم و نتیجه حاصل را داخل پشته قرار می دهیم
- ۴- در پایان حاصل عبارت داخل پشته قرار می گیرد

۵ ۲ ۳ * +

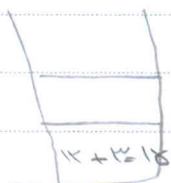
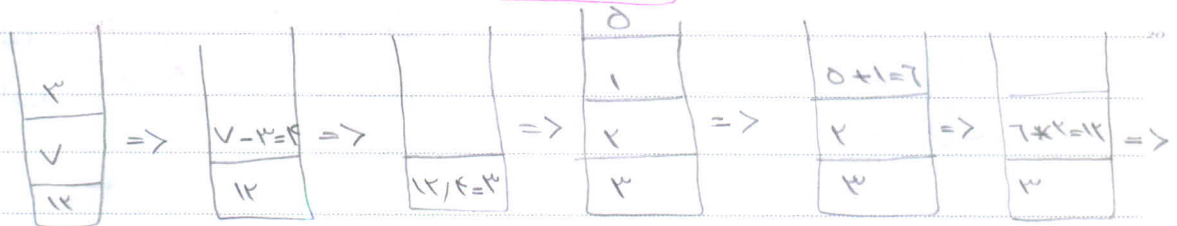
معادله Post Fix را ارزیابی کنید

عملنده ها به صورت عدد اول
جای خود می گذارند

۸ ۳ ۴ + *



۱۲ ۷ ۳ - / ۲ ۱ ۵ + * +



$$3 \quad a \ b / c - d e * + a e * -$$

بدانته‌ها را بهی‌طریق

در جواب 3

inFix $\rightarrow$ preFix

به همین ترتیب به روش پیشوند هر عمل قبل از عملوند ظاهر می‌شود روش دستی

تبدیل inFix $\rightarrow$ preFix با این تفاوت که در مرحله دوم هر عمل را به سمت

چپ بدانته‌ها را خود روش ~~خود~~ منتقل می‌کنیم

$$1 \quad [(((a/b) - c) + (d * e)) - (a * e)]$$

قدم اول

همه را در

$$2 \quad [(((((a/b) - c) + (d * e)) - (a * e))$$

$$3 \quad - + - / a b c * d e * a e$$

تمرین معادل post Fix این عبارت چیست inFix $\rightarrow$ post Fix

$$x + y / (u - t * w) * y$$

$$1 \quad [x + ((y / (u - (t * w))) * y)]$$

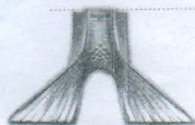
همیشه بدانته

از همه اولی

جایزه است

$$2 \quad [x + (((y / (u - (t * w))) * y)]$$

$$3 \quad x \ y \ u \ t \ w \ * \ - \ / \ y \ * \ +$$



IRAN

تکونی

عبارت چگونگی و چینی و نشر معادل و از یکدایم است

$$a/b - c + b * e - a * c / d$$

چگونگی

$$\left[\left(\left((a/b) - c \right) + (b * e) \right) - (a * (c/d)) \right]$$

$$a/b/c - b e * + a c d / * -$$

پیشوندی

$$\left[\left(\left((a/b) - c \right) + (b * e) \right) - (a * (c/d)) \right]$$

$$- + - / a b c * b e * a / c d$$

15 روشی دلیله عبارت پیشوندی به معادل پسوندی اینها infix و postfix

۱- ابتدا عبارت میان و نشر را حرف به حرف مورد بررسی قدری دهیم تک تک جدا می کنیم

۲- اگر حرف مورد نظر یک عملوند بعد از آن را در عبارت چگونگی می نویسیم

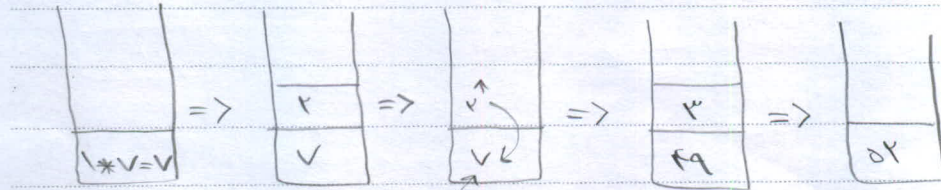
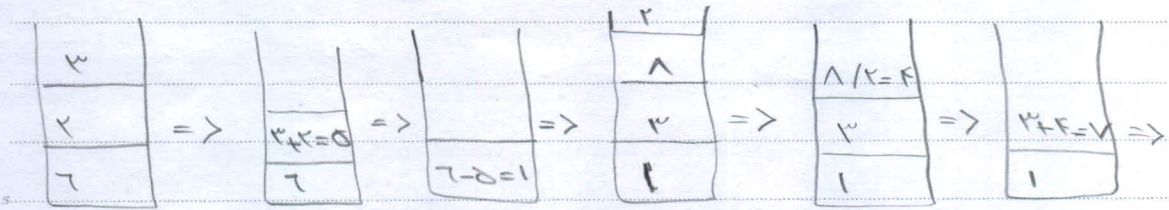
۳- اگر حرف مورد نظر یک عملگر بود آن را بررسی می کنیم اگر داخل پرچمه یا جادار است که از قبل عملگر وجود داشته باشد در این صورت عملگر داخل پرچمه را با عملگر خارج پرچمه مقایسه می کنیم چنانچه عملگر پیشتر آن را از پرچمه خارج کرده و پسوند عملگر را به پرچمه اضافه می نمایم

۴- اگر حرف مورد نظر پرانتز باز بود آن را حتما به داخل پرچمه اضافه می کنیم

۵- اگر حرف مورد نظر پرانتز بسته بود هم به داخل پرچمه قرار دارد و عبارت چگونگی منتقل می کنیم (هنگام پرانتز بسته به پرچمه اضافه نکنیم) در اصل در این حالت هر دو پرانتز باز و بسته را دور می اندازیم در انتهای کار مقننات داخل پرچمه به عبارت چگونگی منتقل می کنیم

۷ ۲ ۳ + - ۳ ۸ ۲ / + * ۲ ↑ ۳ +

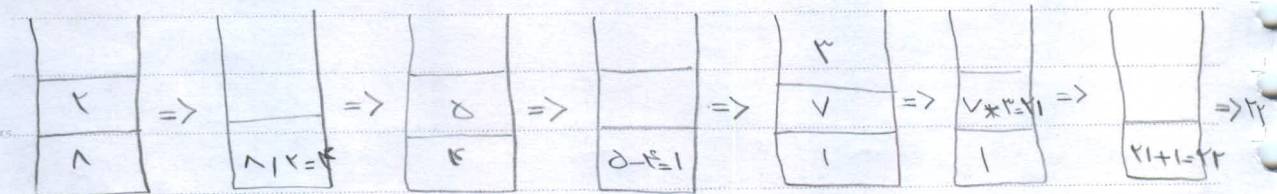
کون
↓



اینجور کون می توانیم ۷ به ۲
یا ۳ به ۴ بدو بکنیم ۳ و ۷

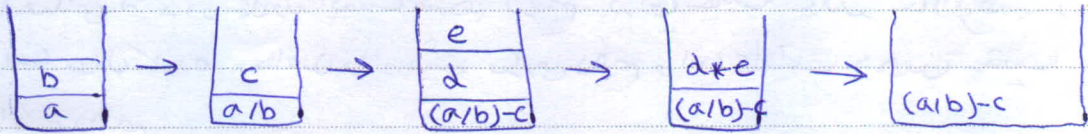
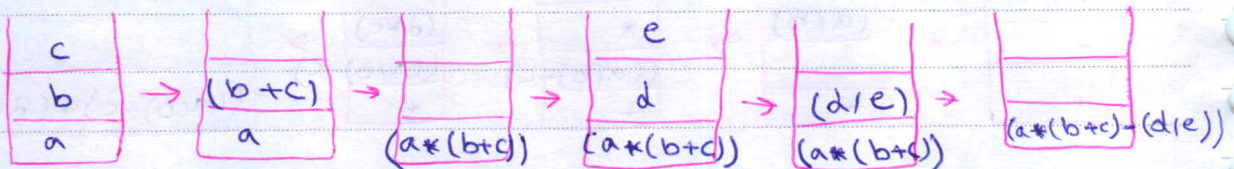
۸ ۲ / ۵ - ۷ ۳ * +

کون



$ab/c - de * +$

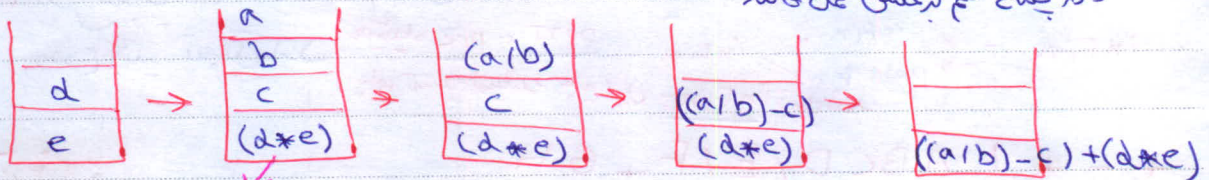
post Fin

 $abc + * de / -$ 

به ترتیب pre Fin و in Fin از همان روش post Fin به این in Fin استفاده کرده با این تفاوت که اینجا از راست به چپ پیمایش می‌کنیم و به گونه‌ای که در تبدیل عبارت مختلف به یکدیگر هیچگاه ترتیباً حروف عملونده‌ها عوض نمی‌شود.

 $+ - / a b c * d e$

در اینجا به عکس عمل می‌کنیم
و در پیش هم به عکس عمل می‌کنیم

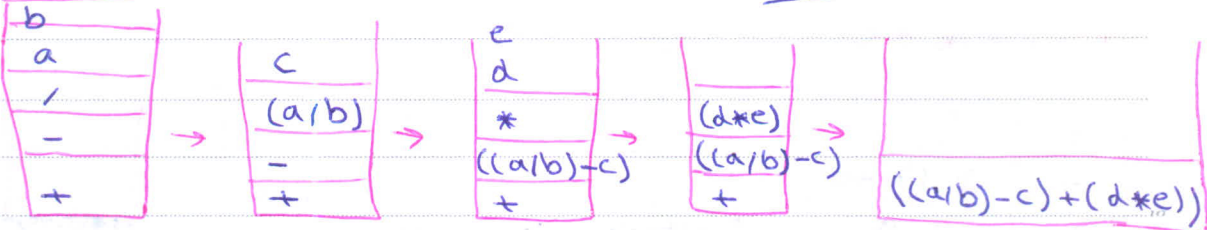


در اینجا هم به عکس عمل
می‌کنیم اول از رو عملونده

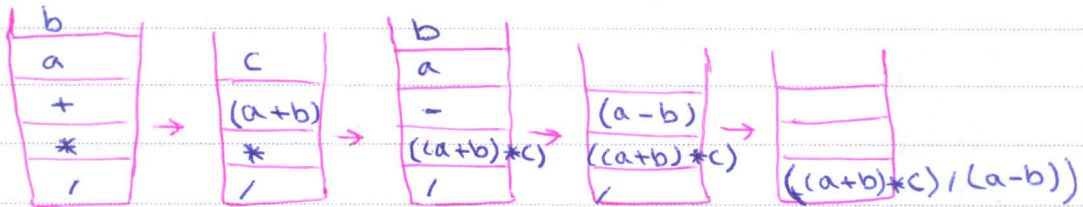
روش اول

در بعضی از کتابها با تبدیل Prefix به in Fix ارائه شده است
 عبارت Prefix را از چپ به راست میخوانی کرده و داده ها را در پشت میزیم
 تا زمانی که به دو عملوند پشت سر هم برسیم در این حالت عملگر جاری می شود
 در اینجا عمل کرده و حاصل را در پشت میزیم قرار می دهیم و این عملیات به همین ترتیب
 ادامه می دهیم

روشی دوم
 $+ - / a b c * d e$

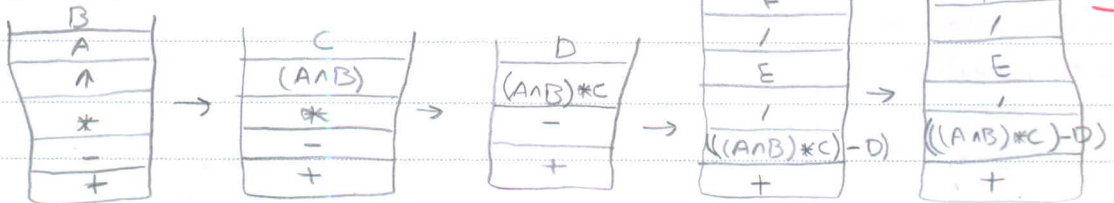


$/ * + a b c - a b$

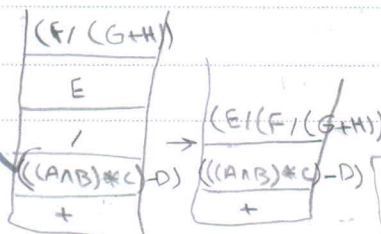


نکته: با تبدیل post Fix به in Fix به بعضی مسائل عبارت in Fix به post Fix تبدیل می شود
 در نوع دیگر تبدیل pre به post به in Fix تبدیل می شود
 به هم تبدیل نمی شود باید اول in Fix تبدیل شود

$+ - * \uparrow A B C D / E / F + G H$



IRAN



$((A^B)*C-D) + ((F/(G+H)) / ((A^B)*C-D))$

$$AB \wedge C * D - EF / GH + / +$$

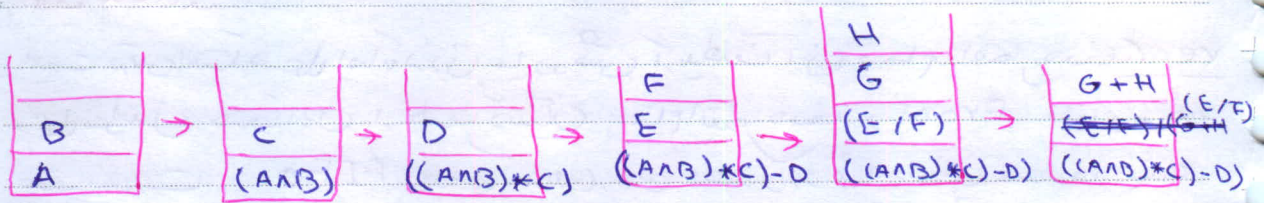
مثال نكته:

عادل عبارت پسوندی است post Fix چون عملها بعد عملوند هستند

هنگامی که post Fix، pre Fix تبدیل شود

تبدیل نمیشود باید in Fix تبدیل شود

1) post Fix $\xrightarrow{\text{تبدیل}}$ in Fix



$$\frac{(E/F) / (G+H)}{((A \wedge B) * C) - D} \xrightarrow{\text{جواب}} (((A \wedge B) * C) - D) + ((E/F) / (G+H))$$

تبدیل به in Fix

2) in Fix $\rightarrow$ pre Fix

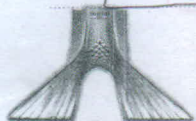
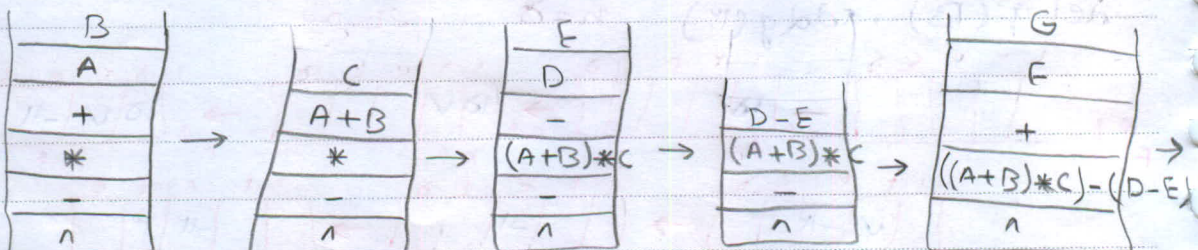
$$(((A \wedge B) * C) - D) + ((E/F) / (G+H))$$

جواب $+ - * \wedge A B C D / / E F + G H$

تبدیل عبارت پیشوندی به پسوندی تبدیل کنیم به پسوندی

pre Fix $\rightarrow$ post Fix

$$\wedge - * + A B C - D E + F G$$



pre Fix $\xrightarrow{\text{تبدیل}}$ in Fix

$$\frac{F+G}{((A \wedge B) * C) - (D-E)}$$

$$(((A+B)*C)-(D-E))^{\wedge}(F+G)$$

in Fin تبدیل

↙

post Fin

$$AB+C*DE--FG+^{\wedge}$$

صف Queue

لیست مرتبی است که عمل اضافه کردن یا زدودن از ابتدای آن به نام انتهای صف rear و عمل خواندن حذف کردن از انتهای آن به نام ابتدای صف و Front انجام می شود. صف به صورت FIFO عمل می کند یعنی اولی عنصر وارد شده اولین عنصر خارج شده می باشد. یکی از کاربردها، مهم حذف در زمان تبلیغ برنامه ها در سیستم عامل می باشد. ساده ترین راه غامبی صف استفاده از یک آرایه یک بعدی به طول n می باشد. باید یک خاصیت مهمی در این آرایه ها احتیاج داریم که Front همیشه به عنصر قبلی از عنصر ابتدای آرایه می کند و دیگری rear که همیشه به عنصر آخر آرایه می کند add q به معنای اضافه کردن به صف و del q به معنای حذف از صف می باشد.

q: Array [1...n] of items

Front, rear : 0...n داده تغییر

Front = rear = 0 مقدار اولیه

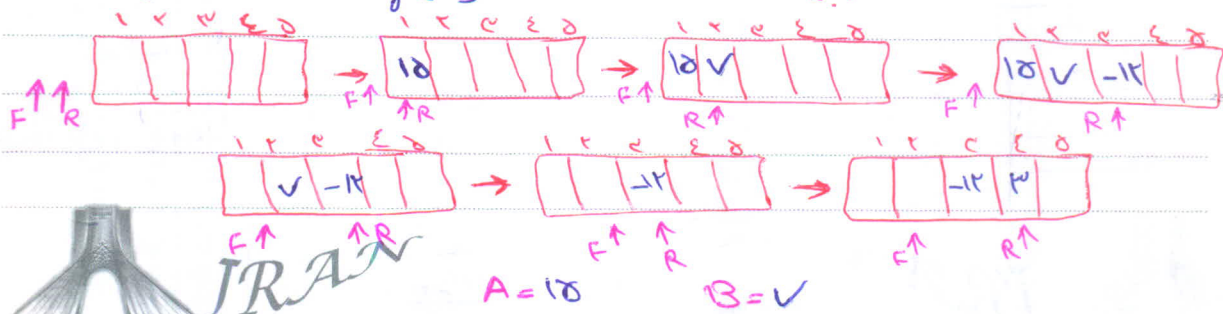
if Front = rear صف خالی بودن

if rear = n شرط پر بودن صف



FR

مثال: عملیات زیر را به ترتیب از چپ به راست به روش کلاسیک نشان دهید. $n=5$
 $add q(10)$, $add q(7)$, $add q(-12)$, $del q(A)$, $del q(B)$, $add q(3)$ پنج خانه $n=5$

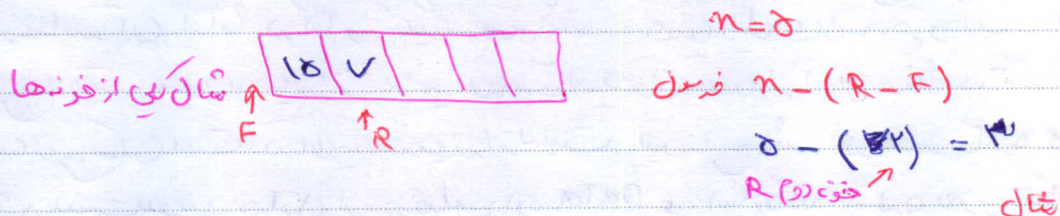


IRAN

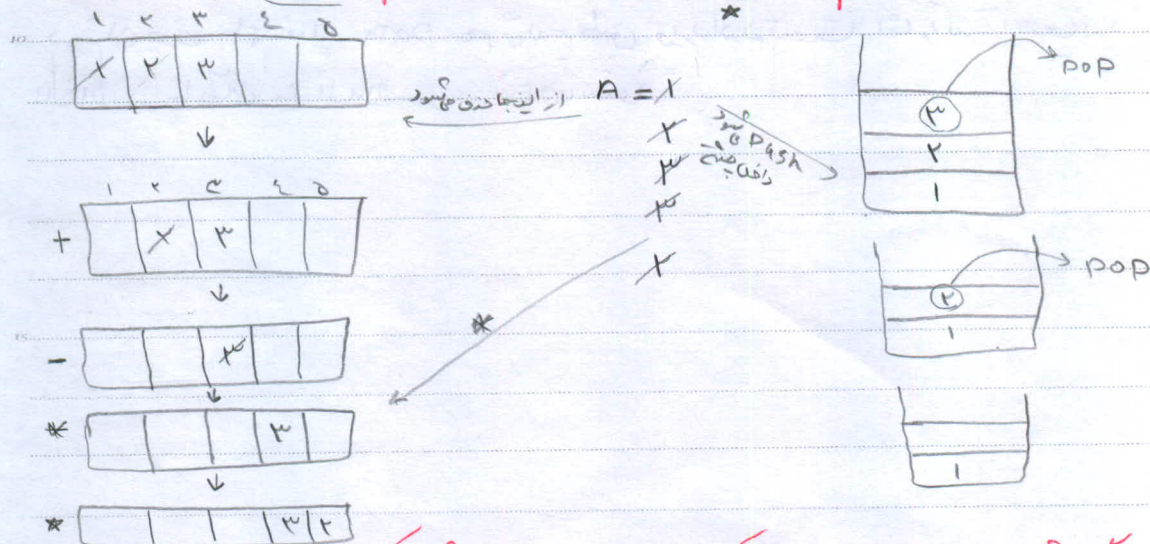
A=10

B=7

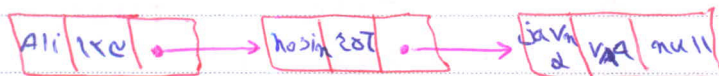
تعداد عناصر موجود در صف معمولی برابر $R - F$ و تعداد خانه‌های خالی $n - (R - F)$ است.
 مشکل اصلی صف معمولی آن است که فقط یکبار قبل استفاده می‌شود و بعداً به آن دسترسی نداریم.
 Rear به انتهای صف اشاره می‌کند و Front به ابتدای آن. $n - (R - F)$



del q(A), push(A), del q(A), push(A), push(A),
 pop(A), add q(A), pop(A), add q(A)



مثالاً برای ذخیره داده‌ها هم می‌توان از آرایه استفاده کرد. هم از لیست پیوندی. عناصر آرایه لزوماً در حافظه پشت سر هم قرار گرفته است ولی عناصر لیست پیوندی از نوع گره‌ها برده و عناصر آن لزوماً در کنار یکدیگر نمی‌باشند. به همین دلیل عمل درج و حذف در لیست پیوندی ساده‌تر و خنثی‌تر است. برای انجام عملی ولی بعضی از اعمال مثلاً جستجو یا مرتب‌سازی ممکن است در آرایه‌ها ساده‌تر از لیست پیوندی باشد. هر عنصر یا گره در node در لیست پیوندی حداقل از دو فیلد داده Data و اشاره‌گر به گره بعدی تشکیل شده است.



در شکل فوق خام فیلد Data هم می‌توانیم طبق وارد در قسمت اشاره‌گر آخری گره که در صف null در پایان است null در نوشته می‌شود.

سطح کُده

اگر ریشه در سطح k در تفرقه n باشد و به اساس تعداد گرهانی که از ریشه فاصله دارند
 به هر سطح می گویند که k هاست. فاصله n از گره ریشه در سطح n قرار دارند
 کده k در سطح k واقع شده است ارتفاع یا عمق درخت

ارتفاع یا عمق درخت

بزرگترین سطح کده ارتفاع یا عمق درخت می نامند
 درخت فوق دارای سطح A می باشد

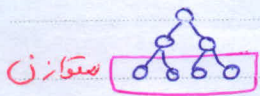
حظی که آره n به کده بعدی رسم شود کده یا یا

شاخه

مسیر از یک برگ ختم شود شاخه خاویه می شود

درخت کتبی

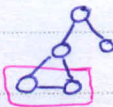
درختی که تعداد فرزندان هر گره در آن حداکثر k باشد در شکل فوق درخت k تایی
 می باشد

درخت متوازن

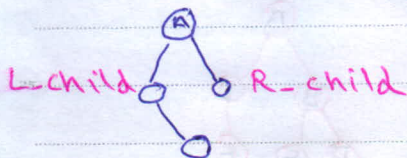
درختی که اختلاف سطح از آن صفر باشد

نکته: در درختها حلقه وجود ندارند

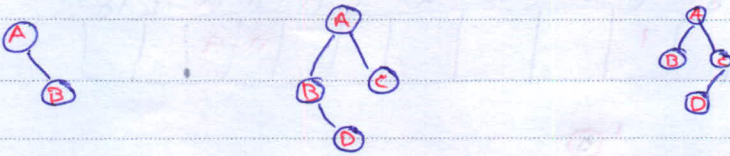
حلقه ایی است

**درخت دویی**

درختی که گره هر گره در فرزند دارد به نام گره های فرزند چپ و راست

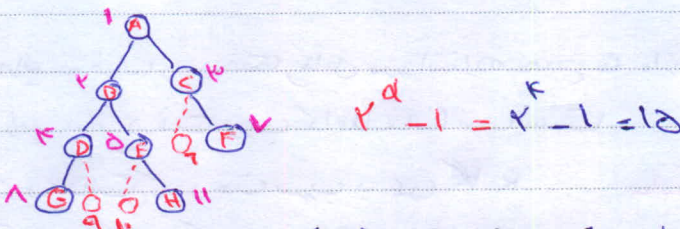


مثال قبلی که درخت دودویی کامل است ولی مثال زیر درخت دودویی کامل نیست



نمایش درخت دودویی با استفاده از آرایه

چون از شماره گذاری آرایه درخت دودویی می‌توان درخت را در آرایه ذخیره کرد بطوریکه به سبب ذخیره درخت آرایه با همان اندیس و شماره ذخیره شود

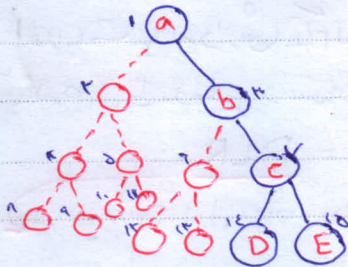


$$2^d - 1 = 2^k - 1 = 15$$

۷ و ۱۵ و ۳۱
و غیره

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D	E		F	G			H				

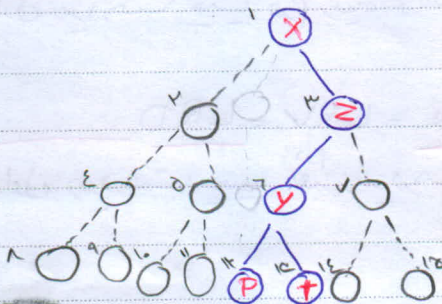
نمایش درختی دودویی آرایه که م خواهد بود



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a		b				c							d	e

همین

آرایه درختی دودویی



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x		z			y					p	q			

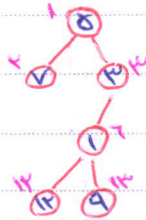


IRAN

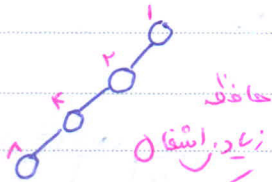
آرایش دایره‌ای ذخیره‌سازی ترتیبی دودویی کامل آن درخت کدام است

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
8	7	3				1					12	9		

حالا شکل درخت را بنویس



استاندارد از فرم آرایه فقط برای درختها کامل مورد استفاده هیچ خانه از حافظه به هدر نمی رود و این فرم آرایه برای درخت مورب کاملاً مناسب است



در واقع در بدنه‌ی حالت یک درخت مورب به یکی 2^k

خانه آرایه نیاز دارد که از این مقدار فقط k محل

اشغال می‌شود که دیگر از این خانه‌ها چیزی آن است نه درج

یا حفظ کرده‌ها مستلزم جایگاهی که حالت خود باعث تغییر شماره سطح نه‌ها می‌شود

اقداماتی که باید داشت عمده‌ها درخت را به یک درخت می‌لینک می‌دهیم درخت می‌خواهیم به ندره فقط یک بار در سه‌ی دایره‌ای با هم به طور کلی ۳ نوع پیماش برای درخت وجود دارد

$preorder = preFin = \underline{NLR} = \underline{VLR} = \underline{DLR}$

در این روش اول ریشه دوم زیر درخت چپ و سوم زیر درخت راست پیماش می‌شود

$inorder = inFin = \underline{LNR} = \underline{LVR} = \underline{LDR}$

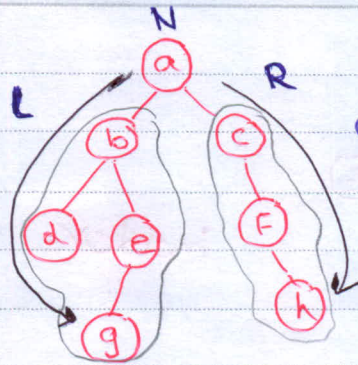
اول زیر درخت چپ دوم ریشه و سوم زیر درخت راست پیماش می‌شود

$postorder = postFin = \underline{LRN} = \underline{LRV} = \underline{LRD}$

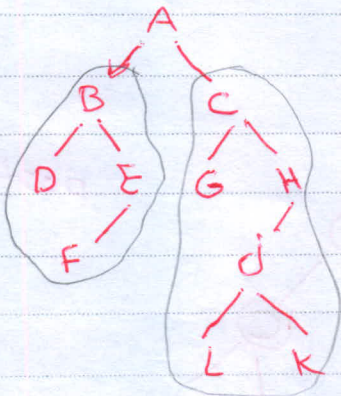
اول زیر درخت چپ بعد زیر درخت راست و سپس ریشه را پیماش می‌شود



preorder = ? NLR
 abdegch



از خود ریشه شروع می شود
 کامل می کند بعد از هر دو زیر درخت

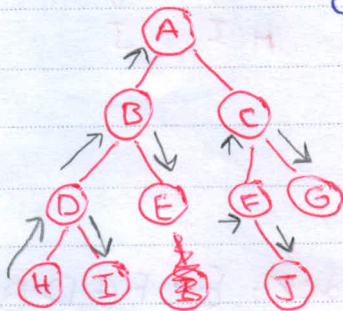


preorder = NLR
 ABDEF C G H J L K

تعیین!

inorder

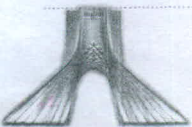
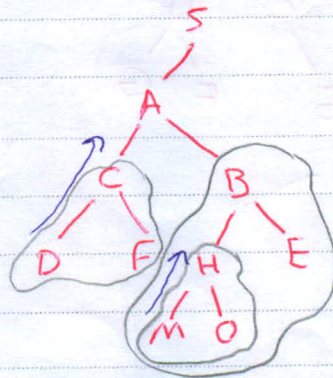
از قسمت چپ می آید یعنی شروع می شود بعد از راست



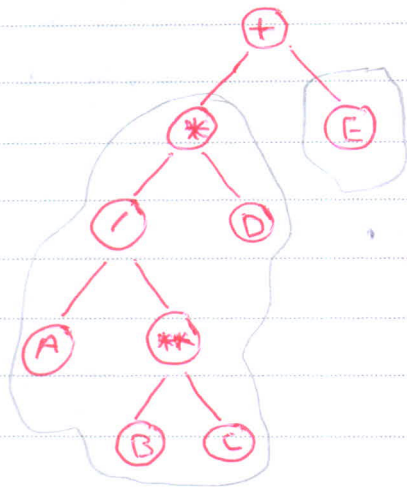
inorder : H D I B E A F J C G

LNR

LNR: DCFAM HOBES



IRAN



inorder = LNR

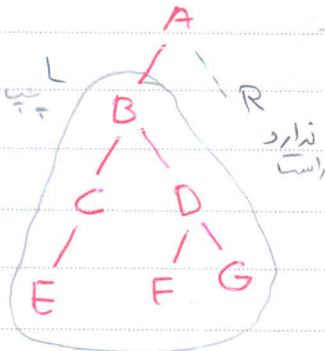
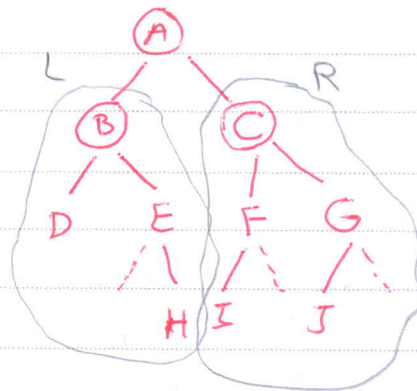
$A/B ** C * D + E$

LRN

postorder:

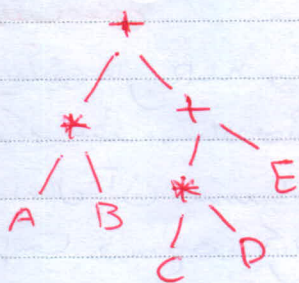
DHEBIFJGCA

postorder



postorder: ECFGDBA

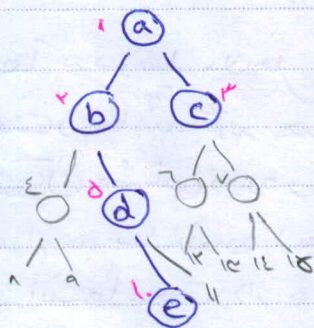
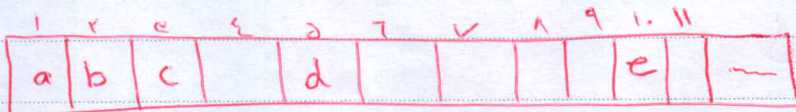
عبدین



postorder: LRN

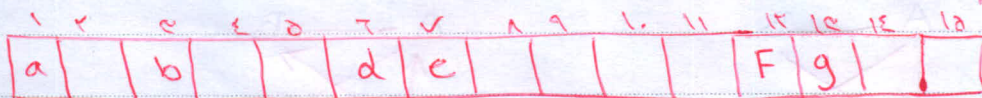
AB * CD * E + +

درخت دودویی T به صورت آرایه نمایش داده شده است. به این روش درخت را به آرایه تبدیل می‌کنیم.



inorder: bedac

ترتیب: inorder و postorder



inorder = LNR

Fdg bea

postorder = LRN

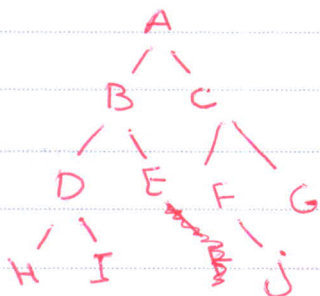
Fgdeba



IRAN

ترتیب دهی

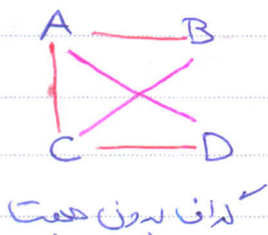
یافتن ها LRN و LNR و NLR در واقع یافش عمقی هستند در بیانی سطحی از یک سطح استفاده می شود در این روش ابتدا ریشه را باز یابی نموده و سپس فرزندان و ... به دنبال آن فرزندان را با یابی می شود این شیوه باز یابی گره ها تنها به نسبت یابی به راست هر سطح جدید یک گره می شود



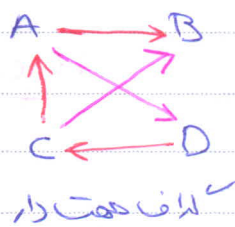
ABCDEFGHIJ

گراف Graph

مجموعه از گره ها که به وسیله ی خطوط به یکدیگر متصل شده است گراف ها می تواند جهت دار یا غیر جهت دار باشد آن که هر میان گراف دارای جهت مشخص باشد بدین معنی که مبدأ یا مقصد آن مشخص بوده و غیر قابل جابجایی است این گراف جهت دار می گویند در گراف جهت دار (نوسان اهمیت) و حال ها از راه یابی ابتدا با آنها ارس می شود و در انت ها جهت دار می توان در هر دو جهت بین اینها حرکت کرد و ترتیب ارس ها آن اهمیت ندارد



گراف بدون جهت



گراف جهت دار

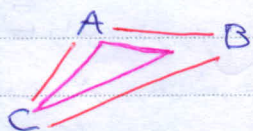
هر گره مستقیماً یا غیر مستقیماً به یک گره متصل باشند را دو گره مجاور یا همسایه می گویند

وزن

خیال‌ها کرده می‌توانند وزن دار یا بدون وزن باشند. کافی که خیال‌ها را از وزن
وزن باشند. لاف وزن دار یا به می‌شود وزن می‌تواند نشان دهنده هزینه
و مسافت یا زمان یا هر مشخصه دیگری از خیال باشد.

حلقه یا Loop

یک حلقه یا Loop است که یک رأس را به خودش متصل می‌کند. به عبارتی دیگر رأس
ابتدا و انتهای یک مسافت است.



خیال‌ها موازی

خیال‌ها موازی یا چندگانه خیال‌ها هستند که رؤس متصل را به هم متصل می‌کند. کافی که در این
خیال‌ها هم متصل باشند. اگر لاف چندگانه می‌گویند.

لاف ساده

لاف بدون خیال موازی و حلقه را لاف ساده می‌گویند. لاف جهت را وقتی ساده می‌گویند
که وقتی موازی نداشته باشد.

نکته: هرآنکه تعداد خیال‌ها لاف جهت دار برابر است با

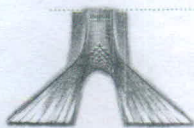
$$n * (n-1)$$

$$\frac{n * (n-1)}{2}$$

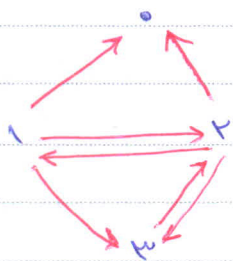
هرآنکه تعداد خیال‌ها لاف بدون جهت دار برابر است با

درجه

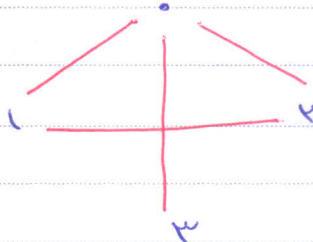
درجه هرگاه تعداد خیال‌ها هرگاه عبور می‌کند در اصل درجه یک رأس تعداد خیال‌ها را متغی
و میان آن رأس است. در لاف جهت دار درجه ورود یک رأس تعداد خیال‌ها
است که به آن رأس وارد می‌شود و درجه خروجی تعداد خیال‌ها است که آن
رأس خارج شده است. اگر درجه ورود یک رأس غیر از میان آن رأس درجه و اگر
درجه خروجی یک رأس نیز باشد آن رأس را زوج می‌گویند.



IRAN

 G_2

مثال:
 عدد ۲ دو تا ورودی
 و ۲ تا خروجی از این روی می‌شود
~~این یک خروجی ندارد~~
~~دو تا ورودی دارد~~

 G_1

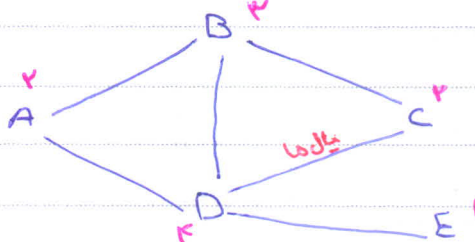
در گراف G_1 رأس ۲ دارای ۳ درجه می‌باشد در گراف G_2 رأس ۲ دارای ۲ درجه دارد و ۲ درجه خارج ۳ می‌باشد کن آن درجه گراف برابر است با درجه گراف که در آن گراف رأس ۲ درجه ۳ می‌باشد

نکته:

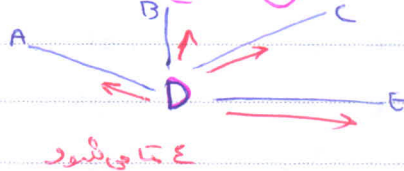
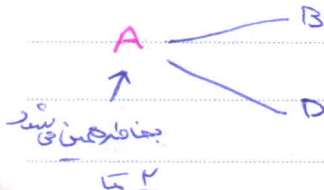
اگر در گراف G با n رأس و $n-1$ و d_i درجه رأس i باشد
 تعداد لبه‌ها از فرمول زیر به دست می‌آید

$$e = \frac{1}{2} \sum_{i=1}^{n-1} d_i$$

$$e = \frac{1}{2} (3 + 2 + 2 + 2 + 1) = 6$$



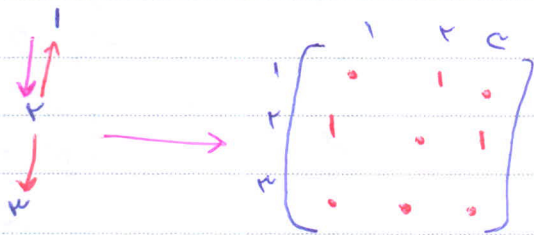
وقتی که تعداد لبه‌ها از یک عدد بزرگتر شود و می‌شود به توان آن به یک عدد مثال



در هر گراف که از رأس هر ستونی در امتداد یک لبه از لبه‌هاست

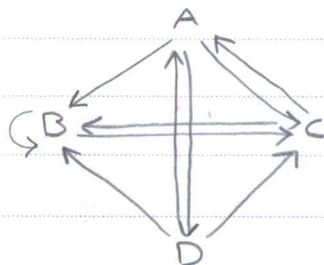
(اس ابتدا یک خیال ها در توانی محسوس می گردد چه طول محسوس تمام خیال ها است
 در یک محسوس طی می شود یک محسوس با طول n دارای $n+1$ یک راس و n خیال
 می باشد در آن وزن دار طول محسوس برابر مجموع وزن ها خیال ها محسوس
 است یک محسوس ساده محسوس است که همه رئوس آن شروع و پایان آن یک راس
 نباشد آنگاه در محسوس یک که به برشته اند یک بار در به شود آن محسوس محسوس ساده
 خواهد بود

نکته: برای اهداف بدون جهت درج هر رأس مانند A مجموع عناصر سطر آن ماتریس است

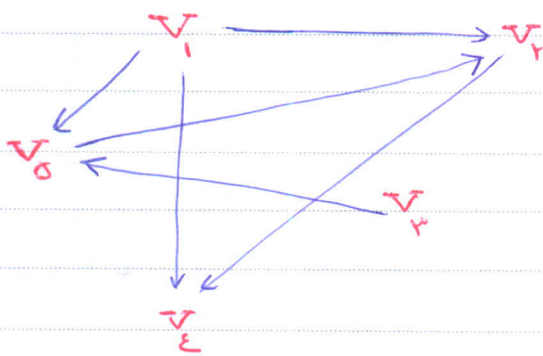


تمرین:

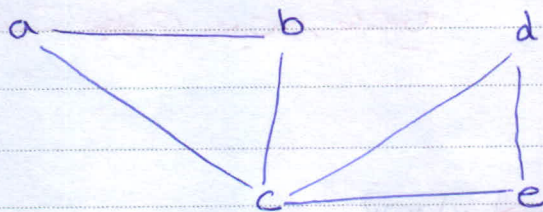
	A	B	C	D
A	.	1	1	1
B	1	.	1	.
C	1	1	.	.
D	1	1	1	.



نکته: مجموع سطر درج خارجی و مجموع سطر درج وارده جدید که خواهد بود در ماتریس هر جواره زیر درج ورودی و خروجی که A را بدست آوریم

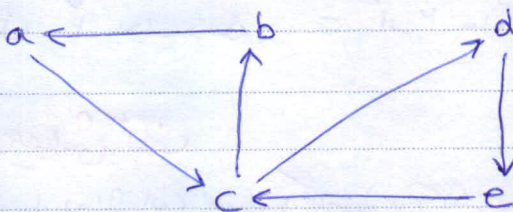


	V_1	V_2	V_3	V_4	V_5	V_6
V_1	.	1	.	1	1	1
V_2	.	.	.	1	.	.
V_3	.	.	.	.	.	1
V_4	.	.	.	.	.	.
V_5	.	1	.	.	.	.



ماتریس

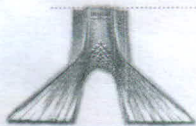
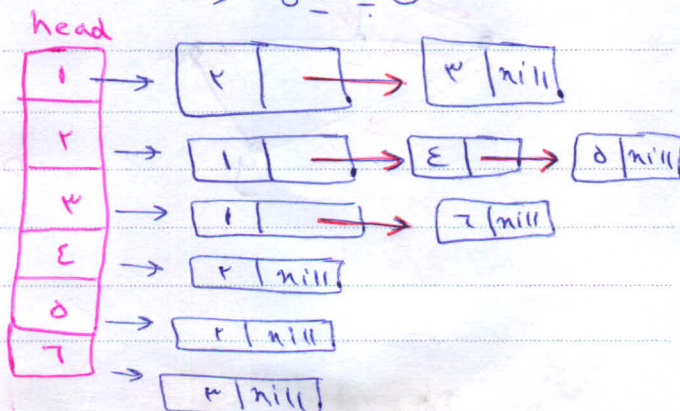
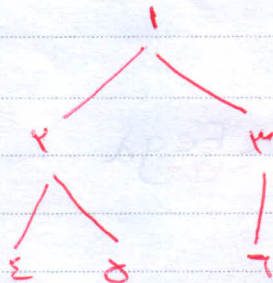
	a	b	c	d	e
a	0	1	1	0	0
b	1	0	1	0	0
c	1	1	0	1	1
d	0	0	1	0	1
e	0	0	1	1	0



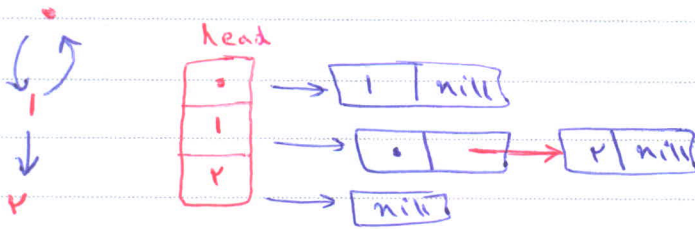
	a	b	c	d	e
a	0	0	1	0	0
b	1	0	0	0	0
c	0	1	0	1	0
d	0	0	0	0	1
e	0	0	1	0	0

لیست مجاورتی

لیست مجاورتی یا هم جوار یکی از ابزارات روش حالت یس مجاورتی آن است که حذف کرده ها در آن مشکل است چرا که بدانکه آن اندازه A تغییر کند و علاوه بر این باید مرتب کند علاوه بر این استوار باشد که باشد آنکه حالت یس A حذف خواهد بود و فضای در حافظه می رود و علاوه بر روش های پیوندی استفاده می کنند این کاری n صفت حالت یس مجاورتی n لیست پیوندی و تری و دیگر راس از گراف می گیر لیست وجود دارد هر راس حداقل ۲ فنل راس و اتصال دارد در لیست منظم حالت آن که ها لیست حاور و راس مجاور از راس می باشد لیست هر راس دارد و مرتب نموده اند این است و این امر دستیابی سریع مجاوره بر راس خاصی به امکان پذیر می سازد



IRAN

لیست مجاورتی جهت دار

چون ۲ بجای وصل نیست آه ۱ جابجا وقت اول وصل می شود

روش پیمایش گراف

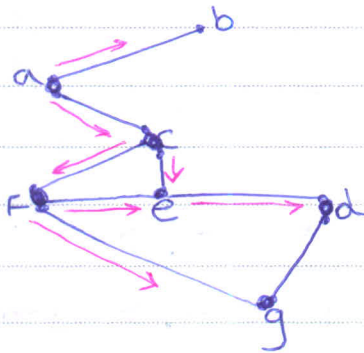
حرف از پیمایش این است کلیه رئوس که از طریق یک رأس قابل دسترس هستند دست آوریم

روش اول سفر BFs : Breadth First search

اول عمق DFS : Deep

جستجوی اول سطح

پیمایش و مجاری BF شروع نموده آن را علامت گذاری می کنیم
اول به یک از رئوس ها از نظر V را در لیست مجاورتی پیمایش نموده زمانی که همه رئوس
مجاور را از V را ملاقات کردیم تمام رئوس ملاقات شده مجاور با اولین
رأس مجاور V در لیست مجاورتی است پیمایش می کنیم اولین سفر تا حدودی شبیه
پیمایش ترتیب سطحی یک درخت است



از همه میوه ها چه در رأس قرار می گیرد مجاورت
استاره می کنیم

BFs : a b c f e g d

جستجو و تحقیق

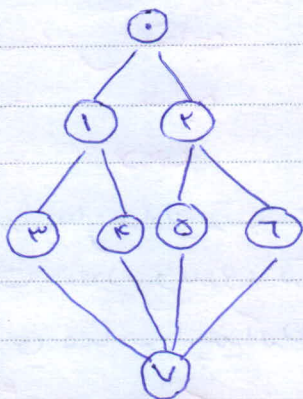
در این روش کار را با داده آغاز می‌کنیم. نگاه کرده در این P که دارد با
V شروع می‌شود و اعداد را می‌کنیم و بی‌همسایه را به دانش خود و نگاه همسایه
البته اگر را به دانش می‌کنیم

بیاض عنق و افزید (اباضوم)

مکملہ روحانی علاج (اس سیرۃ کے ۵۰ بابوں میں)

۱۰ الہ خدیف بید خود من امل علی اللہ انزعی کیم

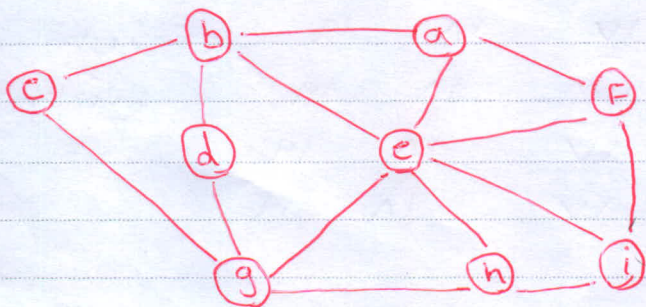
dfs: 1 3 4 5 7



ابتداءً که ها شماره و ترتیبی دهیم از بالا به پایین و از چپ به راست سپس که
فقط اصلاحات خود و در میان خود بدیده می زنیم بعد از آن به سراغ همسایه ها می
رویم و کتبی می رویم یعنی به سراغ که ۱ رفته ایم خود و در میان آن بدیده می زنیم
و سپس به سراغ همسایه ۱ رفته و کتبی را به می زنیم و سپس به سراغ
که ۲ می رویم و سپس سراغ که ۳ و سپس که ۴ می رویم و کتبی که در آن که ۴
هستیم مشاهده می شود که همسایه آن به می زنیم است لذا که در کتبی عقب
به می زنیم و در میان که ۵ و ۶ ای ایستیم و سراغ که ۵ و ۶ که ۲ و سپس که ۷

२१/७

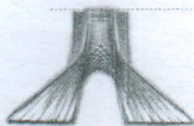
تمہاری



bFs, dFs

bfs: a b e f c d g i h

dfs: a b c g d e f i h



IRAN

مرتبه ساز

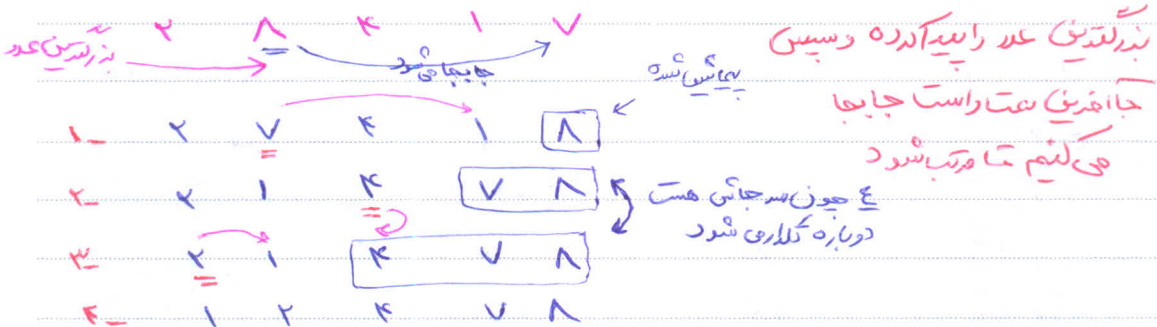
الگوریتم مختلفی برای مرتبه ساز ۸ عنصر وجود دارد که می‌توان آن‌ها را از جهت توانایی مورد بررسی قرار داد هر یک از الگوریتم‌ها بر اساس خواص خود ممکن است در حسابی ویژه کنار گذاشته شود یا نه.

۱- مرتبه ساز انتخابی

۲- حسابی

۳- درجی

۱- یعنی از روی هر اولی مرتبه ساز مقایسه عناصر است این الگوریتم طی چند مرحله عناصر لیست را به صورت صعودی و نزولی مرتب می‌کند در هر مرحله ما بررسی عناصر خاص مرتب می‌کنیم و کلماتی را که در انتهای لیست مشتق می‌کنیم



تجدید

۹	۱۸	۱۰	۱۷	۱۳	۱۹
۹	۱۸	۱۰	۱۷	۱۳	۱۹
۹	۱۳	۱۰	۱۷	۱۸	۱۹
۹	۱۳	۱۰	۱۷	۱۸	۱۹
۹	۱۰	۱۳	۱۷	۱۸	۱۹
۹	۱۰	۱۳	۱۷	۱۸	۱۹

در هر مرحله بدلتنی

را به انتخابی راست

جای جای کنیم

Subject:

Year: Month: Date: ()

۱. ترتیب ساز و جابجایی

باید چندین بار آرایه را به پیش کرد و جابجایی را با عنصر بعدی (در این مثال) خود در
 قدرتی که عنصر اول از عنصر دوم بزرگتر باشد ترتیب ساز صعودی را به اجرا درآورد
 که در مرحله اول n عمل مقایسه انجام می‌گردد و جابجایی داریم

۱۲	۱۸	۱۰	۱۷	۱۳	۹	
۱۲	۱۸	۱۰	۱۷	۱۳	۹	
۱۲	۱۰	۱۸	۱۷	۱۳	۹	
۱۲	۱۰	۱۷	۱۸	۱۳	۹	
۱۲	۱۰	۱۷	۱۳	۱۸	۹	
۱۲	۱۰	۱۷	۱۳	۹	۱۸	

مرحله اول

۱۲	۱۰	۱۷	۱۳	۹	۱۸	
۱۰	۱۲	۱۷	۱۳	۹	۱۸	
۱۰	۱۲	۱۷	۱۳	۹	۱۸	
۱۰	۱۲	۱۳	۱۷	۹	۱۸	
۱۰	۱۲	۱۳	۹	۱۷	۱۸	

مرحله دوم

۱۰	۱۲	۱۵	۹	۱۷	۱۸	
۱۰	۱۲	۱۵	۹	۱۷	۱۸	
۱۰	۱۲	۱۵	۹	۱۷	۱۸	
۱۰	۱۲	۹	۱۵	۱۷	۱۸	
۱۰	۱۲	۹	۱۵	۱۷	۱۸	

مرحله سوم

۱۰	۱۲	۹	۱۵	۱۷	۱۸	
۱۰	۱۲	۹	۱۵	۱۷	۱۸	
۱۰	۹	۱۲	۱۳	۱۷	۱۸	

مرحله چهارم

Subject:

Year..... Month..... Date..... ()

1. 9 12 15 17 18
9 1. 12 15 17 18

مرحله پنجم

4 3 8 1 7 2

مرحله اول

2 4 1 7 2
4 4 1 7 2
4 4 1 7 2
4 4 1 7 2
4 4 1 7 2

مرحله اول

4 4 1 7 2 8
4 4 1 7 2 8
4 1 4 7 2 8
4 1 4 7 2 8
4 1 4 7 2 8

مرحله دوم

4 1 4 2 7 8
1 4 4 2 7 8
1 4 4 2 7 8
1 4 4 2 7 8

مرحله سوم

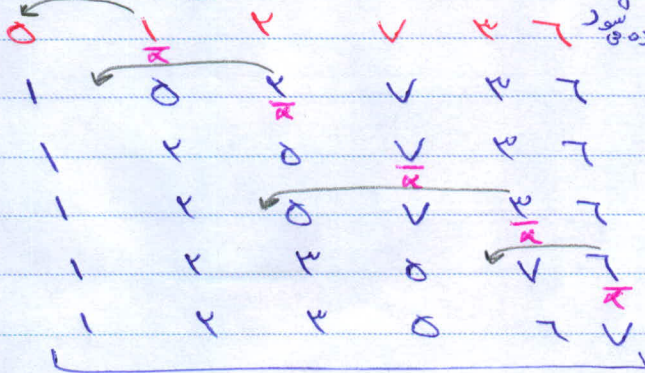
1 4 4 4 7 8
1 4 4 4 7 8
1 2 4 4 7 8

مرحله چهارم

1 2 4 4 7 8
1 2 4 4 7 8

مرحله پنجم

۳ مرتبه سازش درجی یکی از روشهای مقدماتی مرتب سازش منتهی به مقایسه کرد مقایسه جابجایی
 (تبدیل بیست مورد درجه قدر دارد این الگوریتم نیز تقریباً مانند الگوریتم هابی عمل می کند جابجایی تفاوت
 که در این روش مقایسه از عنصر دوم شروع می شود این الگوریتم را با هم مقایسه نموده در صورت نیاز مرتب
 می شود و سپس سوئی عنصر اول و دوم مقایسه شده در صورتی که عنصر سوم از اولی کوچکتر باشد جابجایی
 اولی عنصر می نشیند و عناصر قبلی به سمت راست هل داده می شود اگر عنصر سوم از اولی بزرگتر و از دومی کوچکتر باشد
 بقیه اعداد هم شده و بقیه به سمت راست هل داده می شود



روشنی عدد را به دانش و سپس

بقیه عدد را هل می دهیم

به جلو و عدد را به جایی

تبدیل می دهیم

تبدیل

۱۲	۱۸	۱۰	۱۷	۱۳	۹
۱۲	۱۸	۱۰	۱۷	۱۳	۹
۱۰	۱۲	۱۸	۱۷	۱۳	۹
۱۰	۱۲	۱۷	۱۸	۱۳	۹
۱۰	۱۲	۱۳	۱۷	۱۸	۹
۹	۱۰	۱۲	۱۳	۱۷	۱۸