دانشکده علوم ریاضی و آمار

مدرس: دکتر مجتبی رفیعی نیمسال اول ۱۴۰۰-۱۴۰۱

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۲۷

نگارنده: فرزانه کافی موسوی

۲۹ آبان ۱۴۰۰

فهرست مطالب

۱ یادآوری

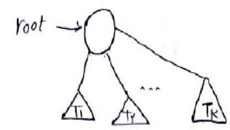
در بخش‌های قبلی دیدیم که روابط بازگشتی یک رویکرد برای حل مساله است که تسهیلاتی برای ما فراهم می‌کندمثل:

- ساده‌تر شدن الگوریتمی که می‌نویسیم
- تحلیل پیچیدگی ساده‌تر به کمک فرم‌هایی که قبلا شروع کردیم

ما می‌توانیم یک خصیه را برای یک ساختار به صورت بازگشتی هم تعریف کنیم.

*** تعریف یک درخت ریشه‌دار**

***تعریف درخت (کاملا) متعادل**



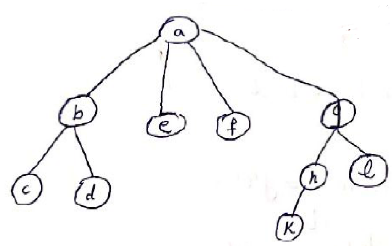
۲ پیمایش درخت‌ها (Traversal tree)

پیمایش درخت‌ها (Traversal tree) می‌توان به ازای نودهای درخت ، پیمایش‌های مختلفی در نظر گرفت. با اینحال پیمایش‌هایی که دارای خواص مطلوب برای کاربردهایی در عمل هستند در دو رده تقسیم‌بندی کرد:

- ترتیب در عمق (Defth–frist)،
- ترتیب در سطح (breadth–first)،

۱.۲ ترتیب در سطح

به این ترتیب است که نودهای یک درخت از ریشه به سمت برگ و از چپ به راست به ترتیب ملاقات می‌شوند، برای نمونه به مثال زیر توجه کنید.



پیمایش سطح $k \quad l \quad h \quad d \quad c \quad g \quad f \quad e \quad b \quad a$

$Tree - Levelorder(T)$
$1. X = t.root$
$2. while(x \neq null) \qquad do$
$3. \qquad visit\ mode\ X \qquad \| visit(x)$
$4. Add\ clildren\ of\ X\ toqueve\ Q \qquad \| left\ to\ right$
$5. X = DeQveve(Q)$

۲.۲ ترتیب در عمق

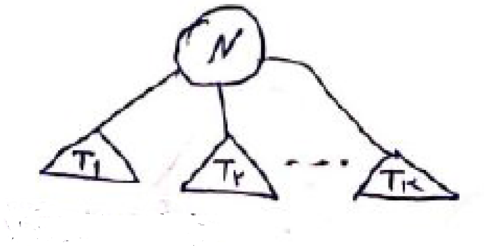
سه پیمایش مطرح در این نوع عبارتند از:

- پیش‌ترتیب(Preorder)
- میان‌ترتیب(Inorder)
- پس‌ترتیب (Postorder)

نکته: در ادامه کلیه پیمایش‌هابرای یک درخت تاییk در نظر گرفته شده است.

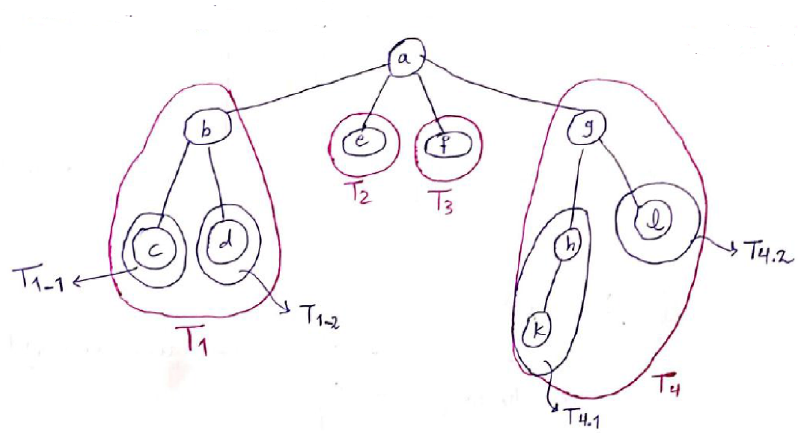
۳.۲ پیمایش پیش‌ترتیب

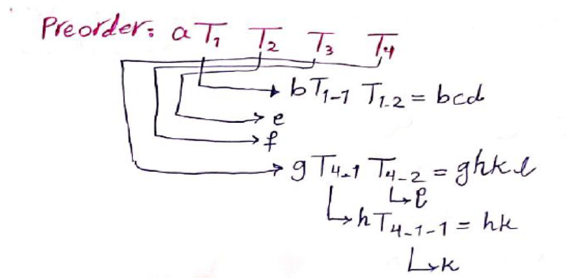
پیمایش پیش‌ترتیب: برای درخت نوعی T ، پیمایش پیش‌ترتیب بصورت زیر است:



$Preorder(T) = N\ T_1\ T_2\ ...T_k$
$Tree - Preorder(T)$
$1.if(T.root \neq null) \qquad then$
$2. \qquad Visit\ node\ X = T.root\ \|Visit(X)$
$3. \qquad Tree - Preorder(T.child_1)$
$\vdots$
$4. \qquad Tree - Preorder(T.child_k)$

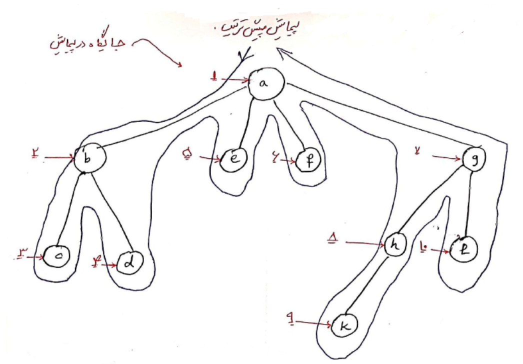
مثال پیمایش پیش‌ترتیب:





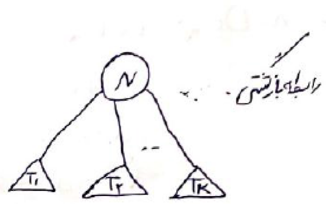
*راه‌حل ساده‌تر برای حل دستی مثال قبل پیمایش پیش‌ترتیب.
پس در نهایت داریم:

$Preorder(T) = a\ b\ c\ d\ e\ f\ g\ h\ k\ l$



۴.۲ پیمایش میان‌ترتیب

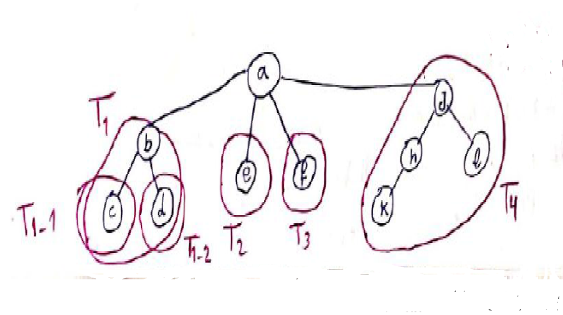
پیمایش میان‌ترتیب: برای درخت چونT ، پیمایش میان‌ترتیب بصورت زیر قابل تعریف است:

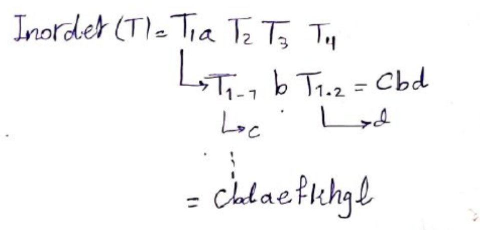


$Inorder(T) = T_1\ N\ T_2\ ...T_k$
$Tree - Inorder(T)$
$1.if(T.root \neq null) \qquad then$
$2.Tree - Inorder(T.child_1)$
$3.visit(T.root)$

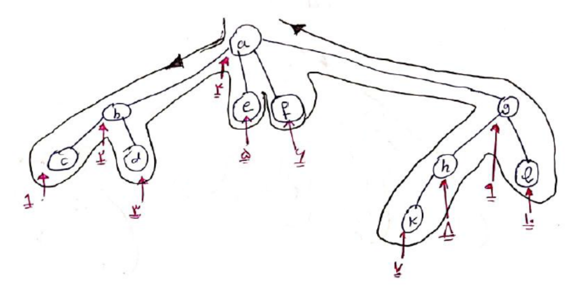
$4.Tree - Inorder(T.child_2)$
$5.Tree - Inorder(T.child_k)$

مثال پیمایش میان‌ترتیب:





راه‌حل دستی- پیمایش میان‌ترتیب:



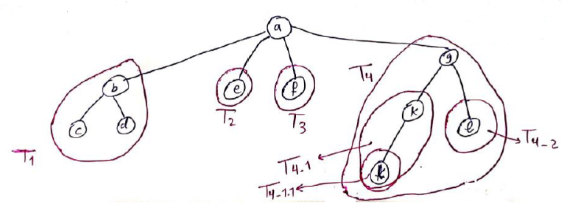
$Inorder(T) = \ d \ o \ d \ a \ e \ f \ k \ h \ g \ l$

۵.۲ پیمایش پس‌ترتیب

پیمایش پس‌ترتیب: برای درخت نوعی چونT ، پیمایش پس‌ترتیب بصورت زیر قابل تعریف است:

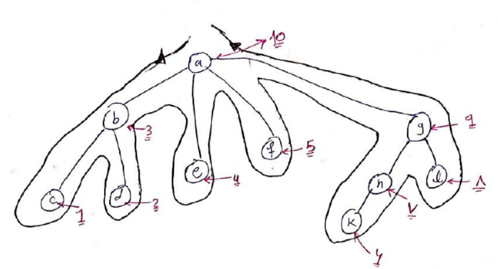
$Postorder(T) = T_1\ T_2\ ...T_k\ N$
$Tree - Postorder(T)$
$1. if (T.root \neq null) \qquad then$
$2.\ Tree - Postorder(T.child_1)$
$3.\ Tree - Postorder(T.child_2)$
$4.\ Tree - Postorder(T.child_k)$
$5.\ visit(T.root)$

مثال پیمایش پس‌ترتیب:



$$Postorder(T) = T_1\ T_2\ T_3\ T_4\ a$$
$$khlg = T_{4-1}\ T_{4.2}\ g$$
$$kh = T_{4-1-1}\ h \qquad l$$
$$= cdbefkhlga$$

راه‌حل دستی- پیمایش پس‌ترتیب:



برخی نکات در رابطه با پیمایش معرفی شده:
۱- از روی هر یک از پیمایش‌های معرفی شده نمی‌توان درخت یکتایی بازسازی کرد، به نمونه‌های زیر دقت کنید.

$*Preorder(T) = a\ b$

$*Inorder(T) = a\ b$

$*Postorder(T) = a\ b$