

پیمایش درخت دودویی

در لیست های خطی میتوان با شروع از ابتدای لیست و حرکت به سمت انتها، داده های لیست را چاپ کرد. با توجه به این که درخت دارای ساختار خطی نیست یافتن الگوریتمی که داده های درخت را به گونه ای چاپ کند که هر داده یکبار و دقیقاً یکبار چاپ شود. از اهمیت بسزایی برخوردار است.

چاپ داده های درخت را پیمایش درخت می گویند. چهار پیمایش زیر را برای درخت دودویی داریم:

- | | |
|----------------------|---------------------|
| ۱. پیمایش میان ترتیب | ۳. پیمایش پیش ترتیب |
| ۲. پیمایش پس ترتیب | ۴. پیمایش سطحی |

پیمایش پیش ترتیب

الگوریتم این پیمایش نیز بازگشتی و بدین صورت است:

۱. ریشه را چاپ کن v
 ۲. زیر درخت چپ را به همین روش پیمایش کن L
 ۳. زیر درخت راست را به همین روش پیمایش کن R
- پیمایش پیش ترتیب به اختصار vLR خوانده می شود.

در واقع پیمایش پیش ترتیب در یک جمله یعنی؛

اولویت با

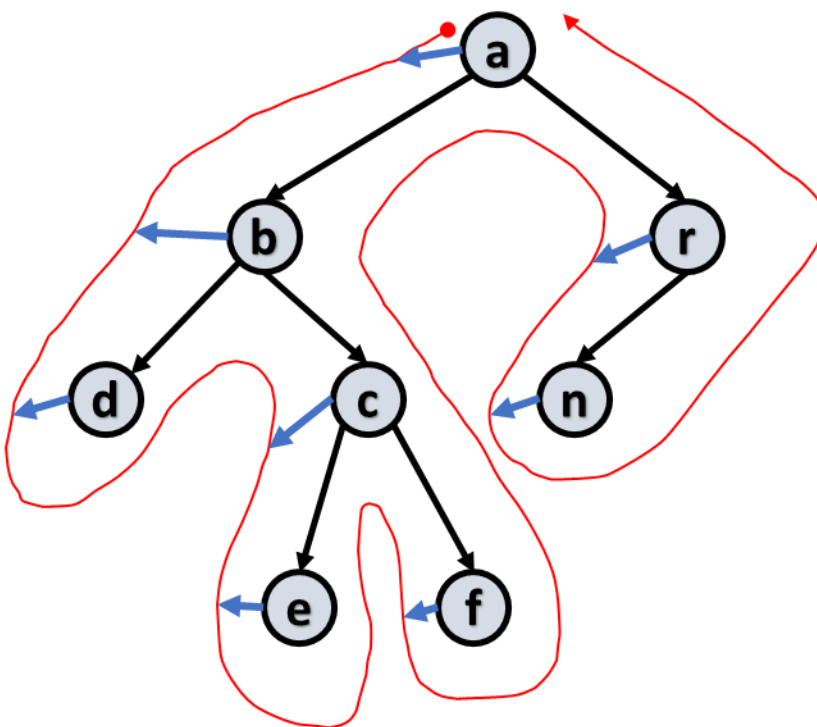
- چاپ ریشه
- گره های زیر درخت چپ
- گره های زیر درخت راست

«پیش» در اسم پیمایش پیش ترتیب اشاره به چاپ ریشه قبل از چاپ داده های زیر درخت چپ و راست دارد.

مثال : این درخت ها را به روش پیش ترتیب پیمایش کنید.

درخت ۲	درخت ۱
<pre> graph TD a((a)) --> b((b)) a --> c((c)) b --> d((d)) d --> m((m)) d --> n((n)) </pre>	<pre> graph TD a((a)) --> b((b)) a --> r((r)) b --> d((d)) b --> c((c)) c --> e((e)) c --> f((f)) r --> n((n)) </pre>
جواب پیش ترتیب برای درخت ۲	جواب پیش ترتیب برای درخت ۱
ABDMNC	ABDCEFRN

جواب پیمایش پیش ترتیب :



پیش ترتیب یک تکنیک حل هم دارد که در آن؛ به دور درخت از چپ به راست یک خط قرمز میکشیم و این خط کشی جریان حرکت پیمایش ماست، نسبت به آنکه میان ترتیب پیمایش میکنیم، پس ترتیب و یا پیش ترتیب بروی این خط داده های گره ها را تصویر میکنیم. برای مثال برای پیش ترتیب که صورت سوال است داده ها را به چپ (پیکان های آبی) مدل میکنیم.

پیمایش میان ترتیب

پیمایش میان ترتیب (جایگاه ریشه) میان مسیر چپ و بعد از زیر درخت چپ و قبل از زیر درخت راست است . الگوریتم این پیمایش نیز بازگشتی و بدین صورت است:

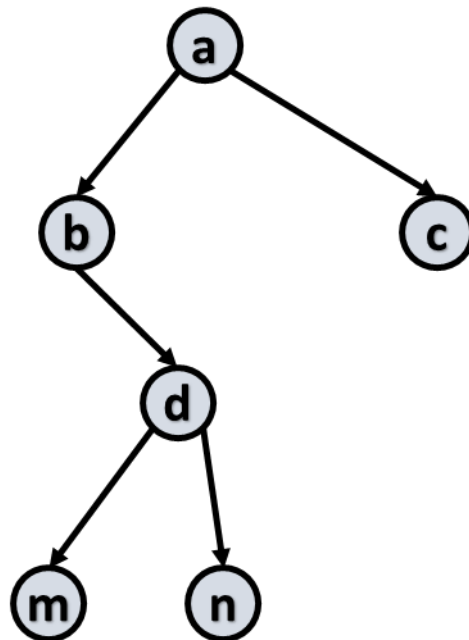
۱. زیر درخت چپ را به همین روش پیمایش کن L

۲. ریشه را چاپ کن v

۳. زیر درخت راست را به همین روش پیمایش کن R

در واقع پیمایش میان ترتیب در یک جمله عبارت است از « ابتدا زیر درخت چپ، بعد از آن ریشه، و در نهایت زیر درخت راست » میان در اسم این پیمایش اشاره به محل چاپ ریشه بعد از چاپ داده های زیر درخت چپ و ثبل از چاپ داده های زیر درخت راست دارد.

درخت ۱

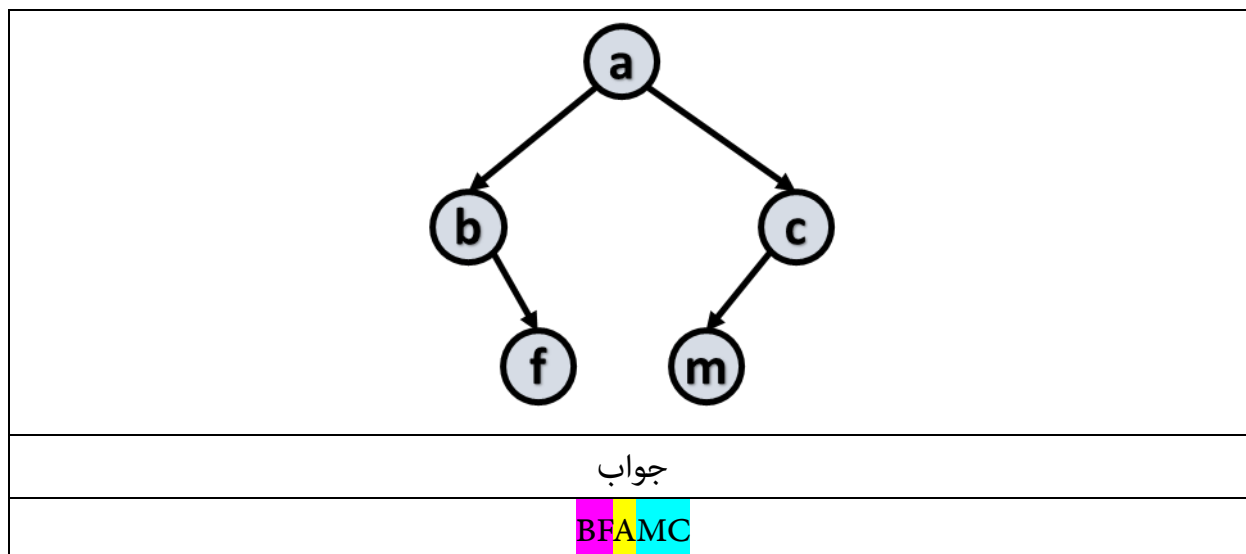


برای چاپ داده های درخت ۱ به صورت میان ترتیب ابتدا باید زیر درخت چپ را به همین روش پیمایش کنیم. زیر درخت چپ خود یک درخت دودویی است که ریشه آن B است. و باید به همین روش (میان ترتیب) داده های آن را چاپ کنیم. برای چاپ داده های این درخت (درخت با ریشه B) ابتدا زیر درخت چپ آن باید پیمایش شود که تهی است و داده ای برای چاپ کردن ندارد. بعد از چاپ زیر درخت چپ خود ریشه B باید چاپ شود. و سپس زیر درخت راست چاپ می شود.

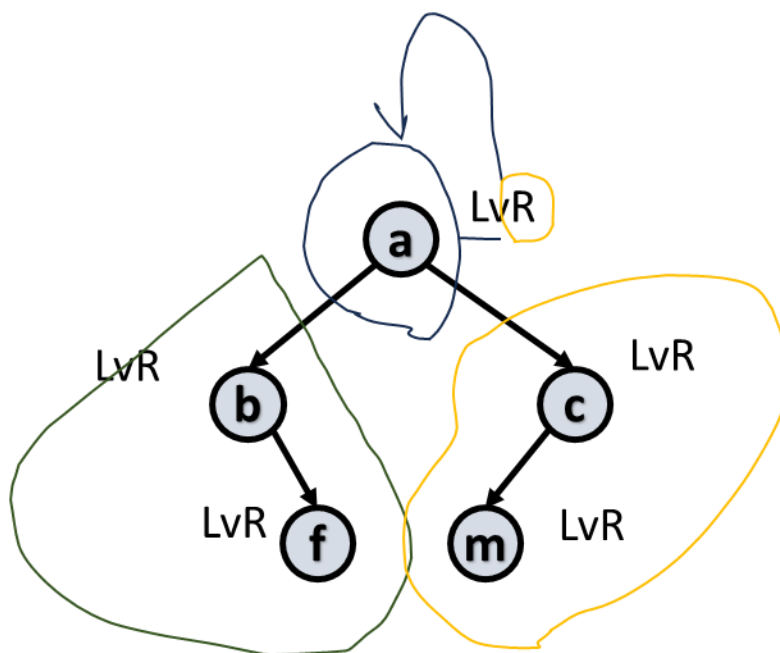
پیمایش میان ترتیب به صورت بازگشتی با تابع زیر قابل پیاده سازی است.

```
1. Inorder(Node t)
2. {
    a. If(t!=null){
        i. Inorder(t.left);
        ii. Print(t.data);
        iii. Inorder(t.right);
    b. }
3. }
```

مثال . این درخت را به صورت میان ترتیب پیمایش کنید.



راهنمای جواب



بالای هر گره LvR را مینویسیم چرا که یکبار باید تابع را برای هر کدام اجرا کنیم.

- از ریشه شروع میکنیم
- به L رسیدیم میریم سراغ زیر درخت چپ و L خط میزنیم

- به v رسیدیم مقدار چاپ میکنیم و v خط میزنیم
- به R رسیدیم میریم زیر درخت راست و R درخت جاری را خط میزنیم.

پیمایش پس ترتیب

الگوریتم این پیمایش نیز بازگشتی و بدین صورت است:

۱. زیر درخت چپ را به همین روش پیمایش کن L
۲. زیر درخت راست را به همین روش پیمایش کن R
۳. ریشه را چاپ کن v

ریشه در آخر، ابتدا نودهای فرزند دیده و چاپ شود. برای پس ترتیب تکنیک های حل میان ترتیب و پس ترتیب کارآمد هست.