



دانشگاه گیلان

دکتر سید ابولقاسم میرروشندل

طراح: تیم حل تمرین DS992

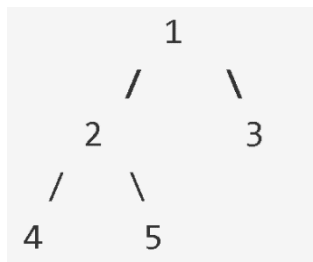
پروژه درس ساختمان های داده (DS992) پیمایش درخت

داستان پروژه :

در این پروژه هدف یادگیری و تسلط بر انواع اعمال قابل پیاده سازی بر روی درخت و پیمایش بر روی آن می باشد. مسئله بدین صورت تعریف میگردد که ورودی به صورت یک درخت دودویی داده میشود. این درخت در برنامه باید به صورت زیر تعریف شده است :

```
class Node{
    int data;
    Node left;
    Node right;
    Node(int data){
        this.data = data;
        left=null;
        right=null;
    }
}
```

یعنی در ورودی آبجکت نود ریشه داده میشود که از آن میتوان به کلیه نود ها (گره های درخت) دسترسی داشت. به طور مثال نود ریشه یا همان نود ۱ در تصویر زیر در ورودی داده میشود و از طریق فرزندان چپ و راست این آبجکت میتوان به نود های دیگر دسترسی پیدا کرد :



هدف پروژه :

هدف از این پروژه ، آشنایی و تسلط دانشجویان در موضوع درخت میباشد.

۱- پیمایش درخت

در ابتدا کلاسی با نام درخت به صورت زیر تعریف کنید.

```
class Tree
{
}
```

حال در این کلاس متدهای زیر را پیاده سازی کنید.

۱-۱ پیمایش میان ترتیب :

این متد در ورودی نود ریشه را دریافت میکند و در خروجی پیمایش میان ترتیب آن را محاسبه می کند. کد Template این متد میتواند به صورت زیر باشد.

```
class Tree
{
    // Return a list containing the inorder traversal of the given tree
    ArrayList<Integer> inOrder(Node root)
    {
        // Code
    }
}
```

۱-۲ پیمایش پیش ترتیب :

مانند قسمت قبل در ورودی نود ریشه را دریافت کرده و پیمایش پیش ترتیب آن درخت را محاسبه نمایید.

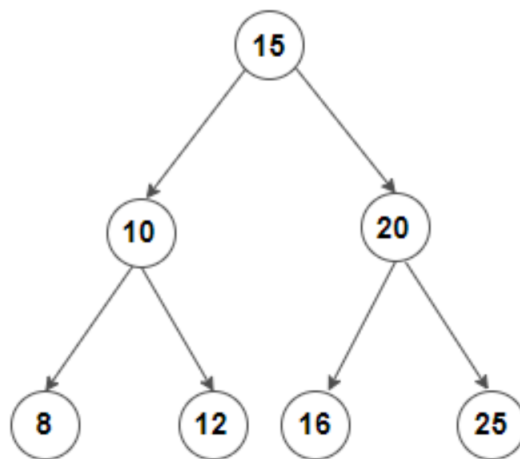
۳-۱ پیمایش پس ترتیب :

متد نوشته شده مانند بخش های قبل پیمایش پس ترتیب را محاسبه نماید.

۲- محاسبه predecessor و successor یک نود در درخت دودویی (نمره اضافی) :

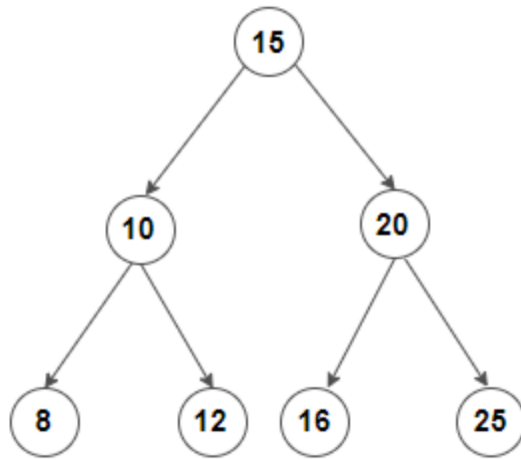
متدی پیاده سازی کنید که predecessor و successor یک نود را در یک درخت دودویی محاسبه نماید. این دو مقدار به صورت زیر محاسبه میگردد.

Successor :



- The inorder successor of 8 is 10
- The inorder successor of 12 is 15
- The inorder successor of 25 does not exist.

Predecessor



The inorder predecessor of 8 does not exist.

The inorder predecessor of 10 is 8

The inorder predecessor of 12 is 10

The inorder predecessor of 20 is 16

۳- محاسبه پیمایش پس ترتیب درخت دودویی از طریق پیمایش میان ترتیب و پیش ترتیب آن (نمره اضافی):

متدی را پیاده سازی نمایید که با گرفتن پیمایش پیش ترتیب و میان ترتیب یک درخت دودویی در ورودی ، پیمایش پس ترتیب را در آن درخت محاسبه کرده و در خروجی نمایش دهد. نمونه ای از ورودی و خروجی این مسئله را در ادامه مشاهده میکنید.

Input:

Inorder traversal in[] = {4, 2, 5, 1, 3, 6}

Preorder traversal pre[] = {1, 2, 4, 5, 3, 6}

Output:

Postorder traversal is {4, 5, 2, 6, 3, 1}

درخت ایجاد شده از این مثال به صورت زیر خواهد بود.

