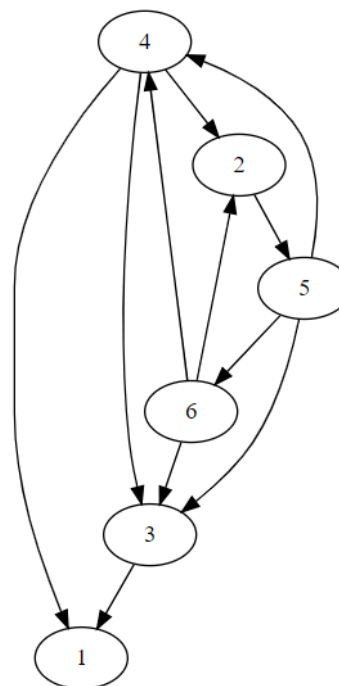


Hint: First consider how to implement access and insert using splay trees. Then think about a special case of reverse in which the $[i, j]$ range is represented by a whole subtree. Use these ideas to solve the real problem. Remember, if you store extra information in the tree, you must state how this information can be maintained under various restructuring operations.

[illegible]

۲. [۲۵] در گراف زیر Strongly-Connected-Components را با نشان دادن مراحل پیدا کنید. توضیح لازم نیست. نشان دادن خروجی الگوریتم در مراحل مختلف لازم است.



۳. [۲۵] با رسم یک گراف G با حداقل ۳ گره به همراه معکوس آن G^R نشان دهید که کوچکترین post-order یا کوچکترین/بزرگترین pre-order در هیچکدام از گراف‌ها نمی‌توان گره Sink در گراف اصلی را پیدا کرد. در هر مثال لازم است شماره‌های pre-order و post-order هر نود مشخص شده و گره Sink و Source نیز مشخص شود. همچنین مشخص کنید که بزرگتر/کوچکتر اعداد کدام گره‌ها هستند. هدف این سوال نشان دادن این است که راه درست پیدا کردن گره Sink در گراف اصلی پیدا کردن گره با بیشترین post-order در گراف معکوس می‌باشد.

۴. [۲۵] ورودی متد زیر یال های یک گراف بدون جهت است. خروجی آن چیست؟ پیچیدگی محاسباتی این متد بر حسب تعداد گره ها N و تعداد یال ها E چیست؟ $O(\text{_____})$ در نقطه چین زیر توضیح دهید.

```
// "edges" contains a list of edges for an undirected
// graph. Each edge is represented by an array of size 2.
// The first element is the source node and the second
// element is the target node.
public static bool Solve(long nodeCount, long[][] edges)
{
    DisjointSet ds = new DisjointSet(edges.Length);
    for(int i=0; i<nodeCount; i++)
        ds.MakeSet(i);

    foreach(var edge in edges)
    {
        var x = ds.Find(edge[0]);
        var y = ds.Find(edge[1]);
        if (x == y)
            return true;
        ds.Union(x, y);
    }
    return false;
}
```

۵. [۲۵] اعداد زیر را به ترتیب در یک Left Leaning Red Black Tree به همراه درخت ۲-۳ متناظر آن اضافه کرده و درخت های نتیجه را رسم کنید. نودهای قرمز را با دو دایره تو در تو مشخص کنید.

۱، ۲، ۴، ۸، ۳، ۷، ۱۲، ۱۸، ۵، ۹، ۱۶