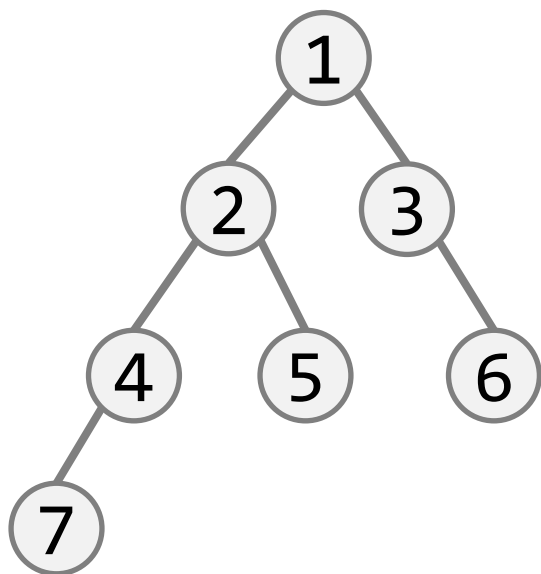


太戈编程  
etiger.vip

# 信奥算法



请写出dfs访问序列

1247536

1254736

1362475

1362547

dfs序列并不唯一

节点的dfs序号

**dfn : depth-first number**

**dfn[u]**表示原u号节点  
在**dfs**访问时第几个被访问到

$\text{dfn}[u]$ 表示原 $u$ 号节点  
在 $\text{dfs}$ 访问时第几个被访问到

已知 $\text{dfs}$ 访问序列

请写出每个点的 $\text{dfn}[]$

1423

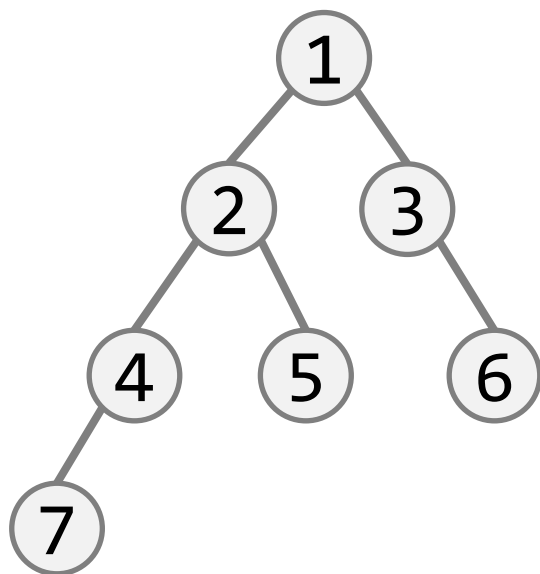
1342

1247536

1263574

52413

42531



dfs序列：1247536

节点的dfn[]：1263574

dfs序列里,祖先总在子孙前

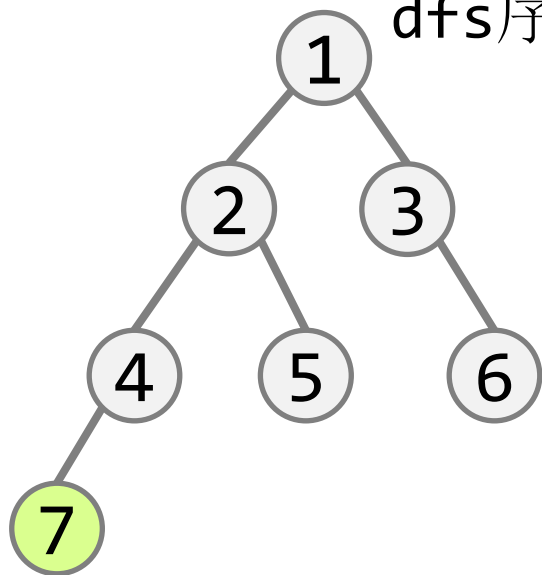
dfs序列里,子树节点总是连续段

请同学发现规律

dfs序列中原节点编号

dfs序列中位置编号

1	2	4	7	5	3	6
1	2	3	4	5	6	7



dfs序列里,子树节点总是连续段

1号为根的子树对应dfs序号[1,7]

2号为根的子树对应dfs序号[2,5]

3号为根的子树对应dfs序号[6,7]

4号为根的子树对应dfs序号[3,4]

5号为根的子树对应dfs序号[5,5]

6号为根的子树对应dfs序号[7,7]

7号为根的子树对应dfs序号[4,4]

dfs序列中原节点编号

1

2

4

7

5

3

6

dfs序号

1

2

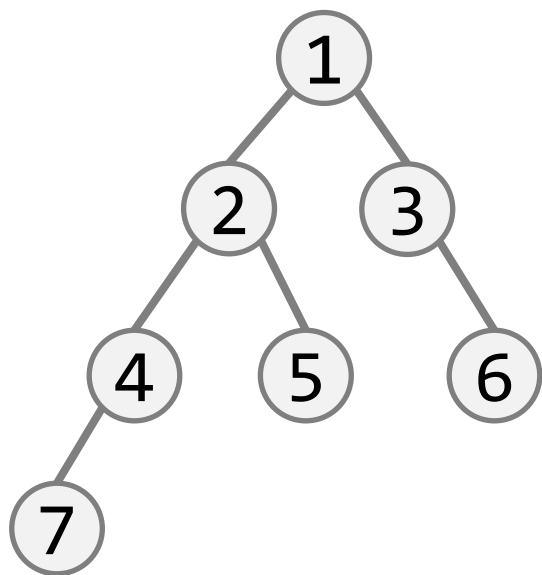
3

4

5

6

7



dfs序列里,子树节点总是连续段

u为根的子树对应dfs序号是个区间

[  $dfn[u]$ ,  $dfn[u]+sz[u]-1$  ]

```
6 struct subTree{int L,R;} T[N];
```

u为根的子树  
对应dfs序号区间  
[ T[u].L,T[u].R ]

```
7 int dfn[N],timer;  
8 void dfs(int u,int fa){  
9     T[u].L=dfn[u]=++timer;  
10    for(int i=0;i<to[u].size();++i)  
11        if(to[u][i]!=fa) dfs(to[u][i],u);  
12    T[u].R=timer;  
13 }
```




判断u是不是v的严格直系祖先

区间  
包含

判断 区间  $[T[u].L, T[u].R]$   
是否包含 区间  $[T[v].L, T[v].R]$

```
14 bool up(int u, int v){  
15     return T[u].L < T[v].L && T[v].R <= T[u].R;  
16 }
```



若  $u == v$   
返回 0

2161

# 解法思路的形成 如何启发思路

**A**

暴力

**B**

简化为链/序列化



## 暴力如何求解

LOSE  $u$   $p$

直接修改  $x[u] -= p$

CHECK  $u$

暴力访问 $u$ 子树

复杂度是多少?

最差情况 $O(nq)$

随机数据的平均情况 $O(q \log n)$

随机树的高度 $O(\log n)$



```
6 void dfs_p(int u,int fa){
7     prnt[u]=fa;
8     for(int i=0;i<to[u].size();++i)
9         if(to[u][i]!=fa) dfs_p(to[u][i],u);
10 }
```

prnt[u]表示u的父节点

```
11 void dfs_sub(int u,int fa){
12     ans+=x[u];
13     for(int i=0;i<to[u].size();++i)
14         if(to[u][i]!=fa) dfs_sub(to[u][i],u);
15 }
```

# 解法思路的形成 如何启发思路

**A**

暴力

**B**

简化为链/序列化



## 树简化为链

请在链状情况重述原题

点修改

后缀和询问

动态rsq问题

用bit维护



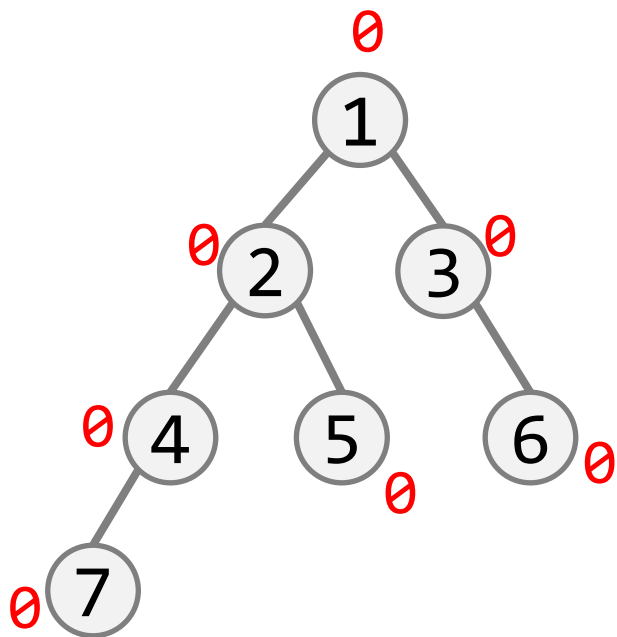
# 树的序列化

将树结构  
转换为一维序列  
使用序列问题的  
经典工具

先求出**dfs**序列相关信息

**dfs序+bit**





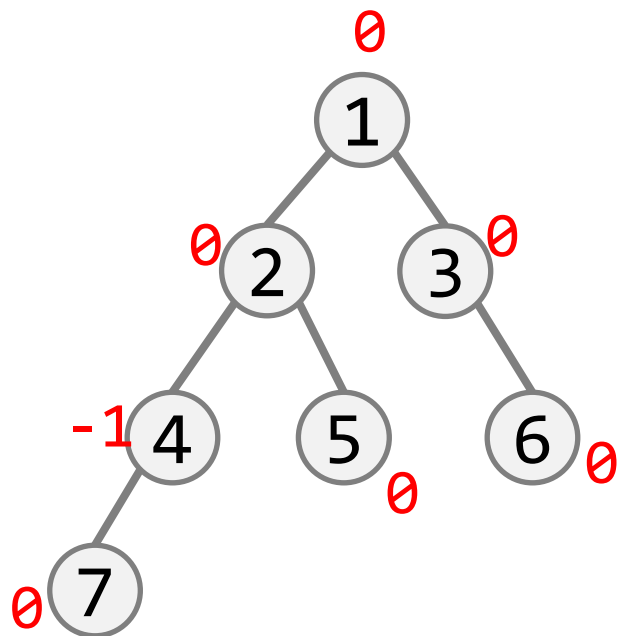
节点编号u

dfn[u]

x[u]

1	2	4	7	5	3	6
1	2	3	4	5	6	7
0	0	0	0	0	0	0

LOSE 4 1



LOSE 4 1

节点编号u

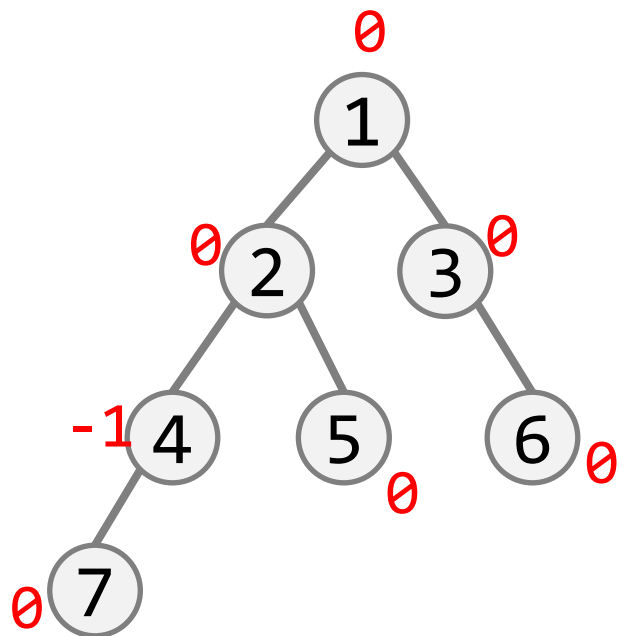
dfn[u]

x[u]

1	2	4	7	5	3	6
1	2	3	4	5	6	7
0	0	-1	0	0	0	0

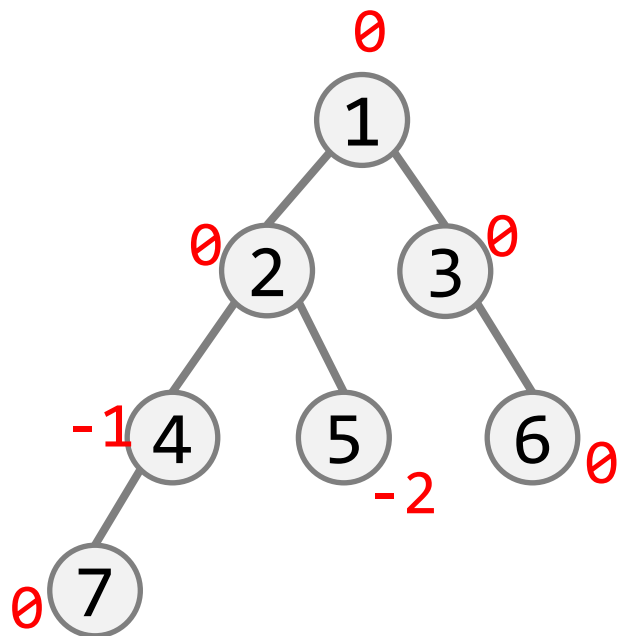
bit操作  
add(3, -1)

dfn[4]



节点编号u	1	2	4	7	5	3	6
dfn[u]	1	2	3	4	5	6	7
x[u]	0	0	-1	0	0	0	0

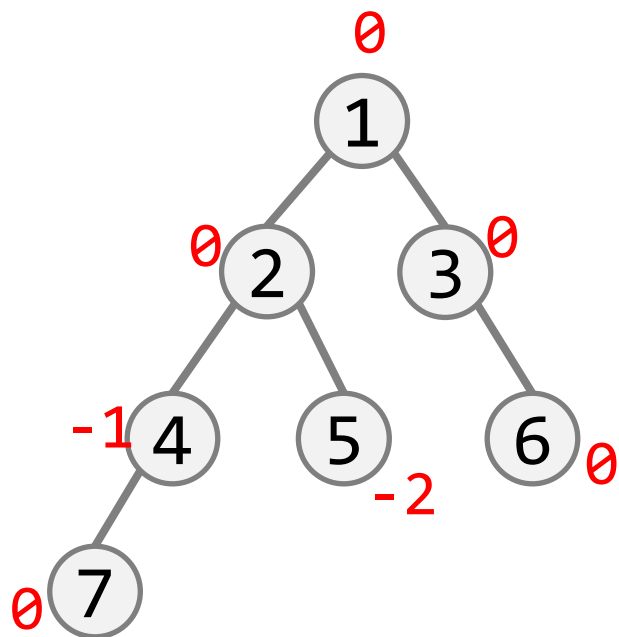
LOSE 5 2



节点编号u	1	2	4	7	5	3	6
dfn[u]	1	2	3	4	5	6	7
x[u]	0	0	-1	0	-2	0	0

LOSE 5 2

bit操作  
add(5, -2)



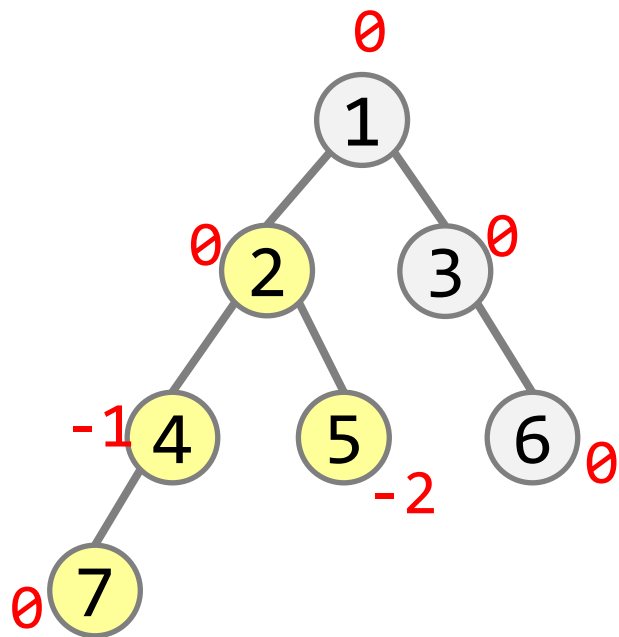
节点编号u

dfn[u]

x[u]

1	2	4	7	5	3	6
1	2	3	4	5	6	7
0	0	-1	0	-2	0	0

CHECK 2



节点编号u

dfn[u]

x[u]

1	2	4	7	5	3	6
1	2	3	4	5	6	7
0	0	-1	0	-2	0	0

CHECK 2

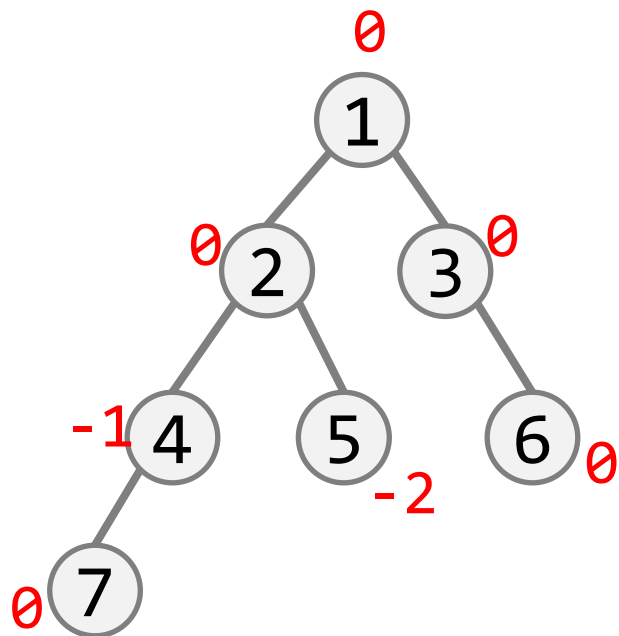
子树对应dfn区间[2,5]

bit操作

$\text{psq}(5) - \text{psq}(2-1)$

$\text{dfn}[2] + \text{sz}[2] - 1$

$\text{dfn}[2] - 1$



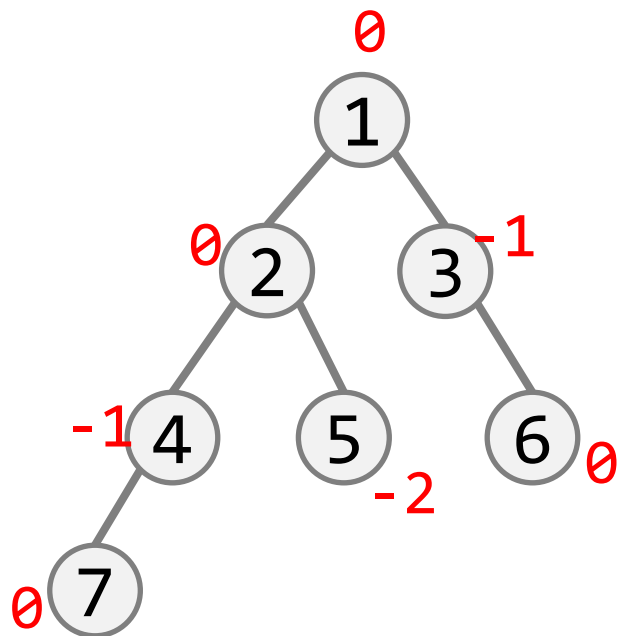
节点编号u

dfn[u]

x[u]

1	2	4	7	5	3	6
1	2	3	4	5	6	7
0	0	-1	0	-2	0	0

LOSE 3 1



LOSE 3 1

节点编号u

dfn[u]

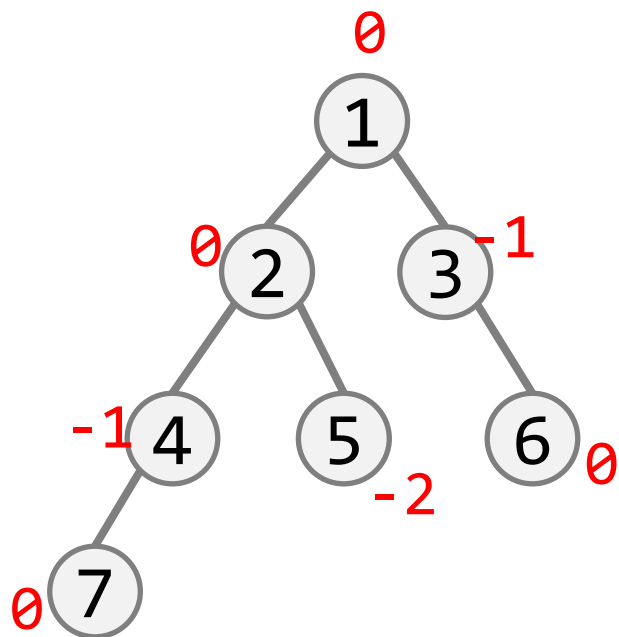
x[u]

1	2	4	7	5	3	6
1	2	3	4	5	6	7
0	0	-1	0	-2	-1	0

bit操作  
add(6, -1)

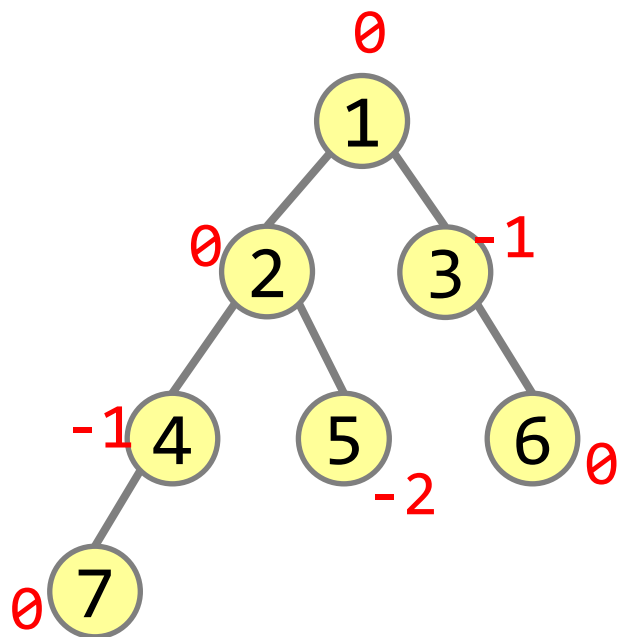
dfn[3]





节点编号u	1	2	4	7	5	3	6
dfn[u]	1	2	3	4	5	6	7
x[u]	0	0	-1	0	-2	-1	0

CHECK 1



CHECK 1

节点编号u	1	2	4	7	5	3	6
dfn[u]	1	2	3	4	5	6	7
x[u]	0	0	-1	0	-2	-1	0

子树对应dfn区间[1,7]
bit操作 psq(7)-psq(1-1)



# 小结

树形问题常规解题思路

树的序列化

+

数据结构



# 树的序列化

serialization

A

dfs序列

B

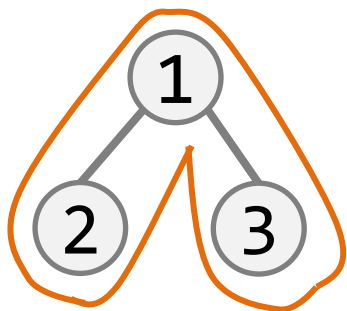
第1类欧拉序列

C

第2类欧拉序列

D

bfs序列

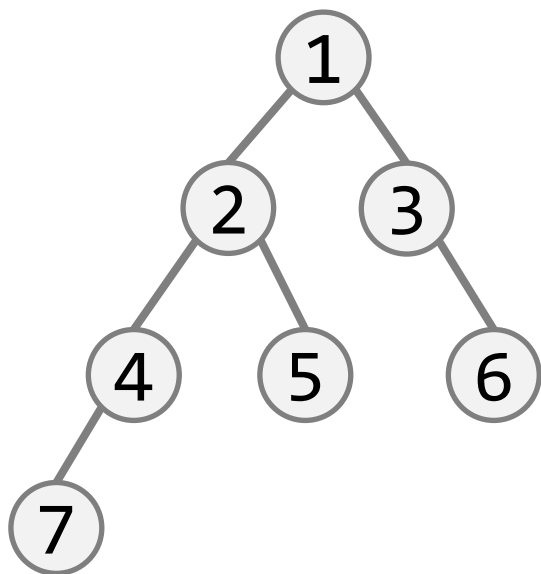


第1类欧拉序列  
**122331**

第2类欧拉序列  
**12131**

欧拉序列

dfs访问+回溯



第1类欧拉序列

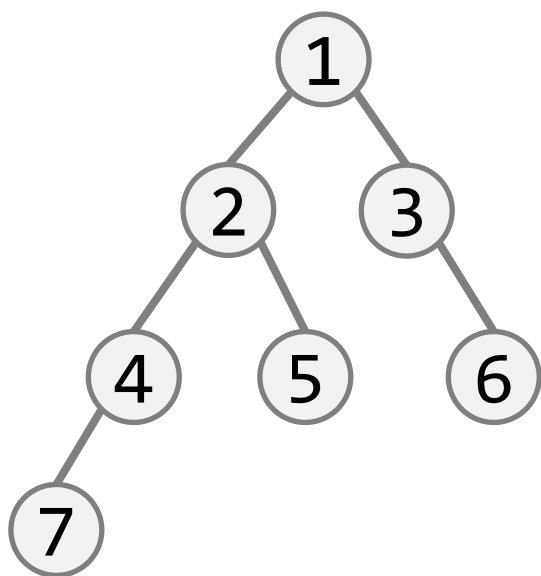
12477455236631

第2类欧拉序列

1247425213631

根节点作隔板分割子树

请同学发现规律



欧拉序列

dfs访问+回溯

共 $n$ 个节点时

第1类欧拉序列多长

$$n*2$$

第2类欧拉序列多长

$$n*2-1$$

易错点

数组大小设置太小



# 信息的保留/丢失

欧拉序列能否复原出树结构?

能

dfs序列能否复原出树结构?

不能

dfs序列+sz[ ]  
可以复原出树结构!





# 太戈编程

1011

2161

2162