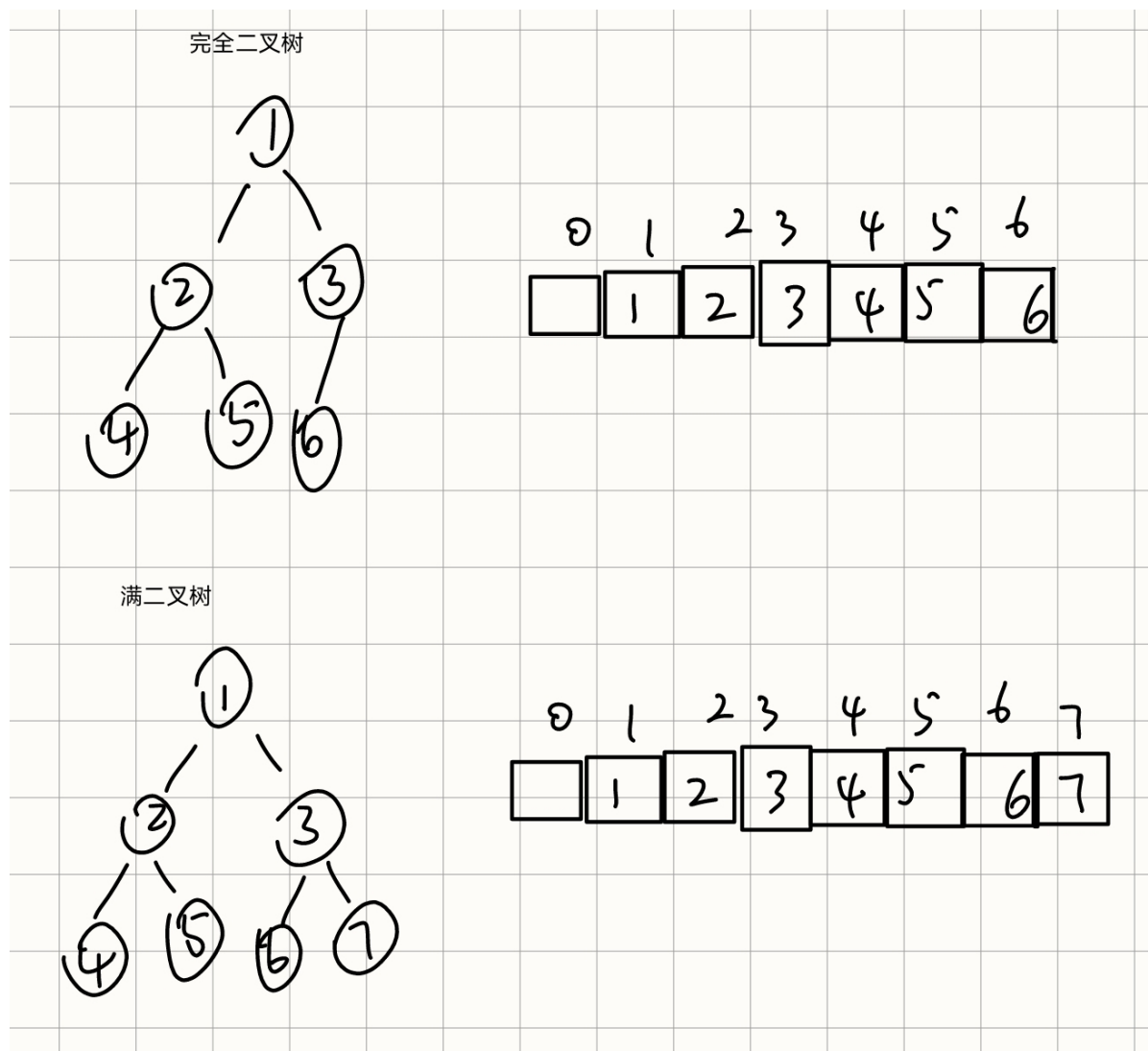


大家好，我是蓝蓝，这是我们一期数据结构应用题专题的第一天。day07/15

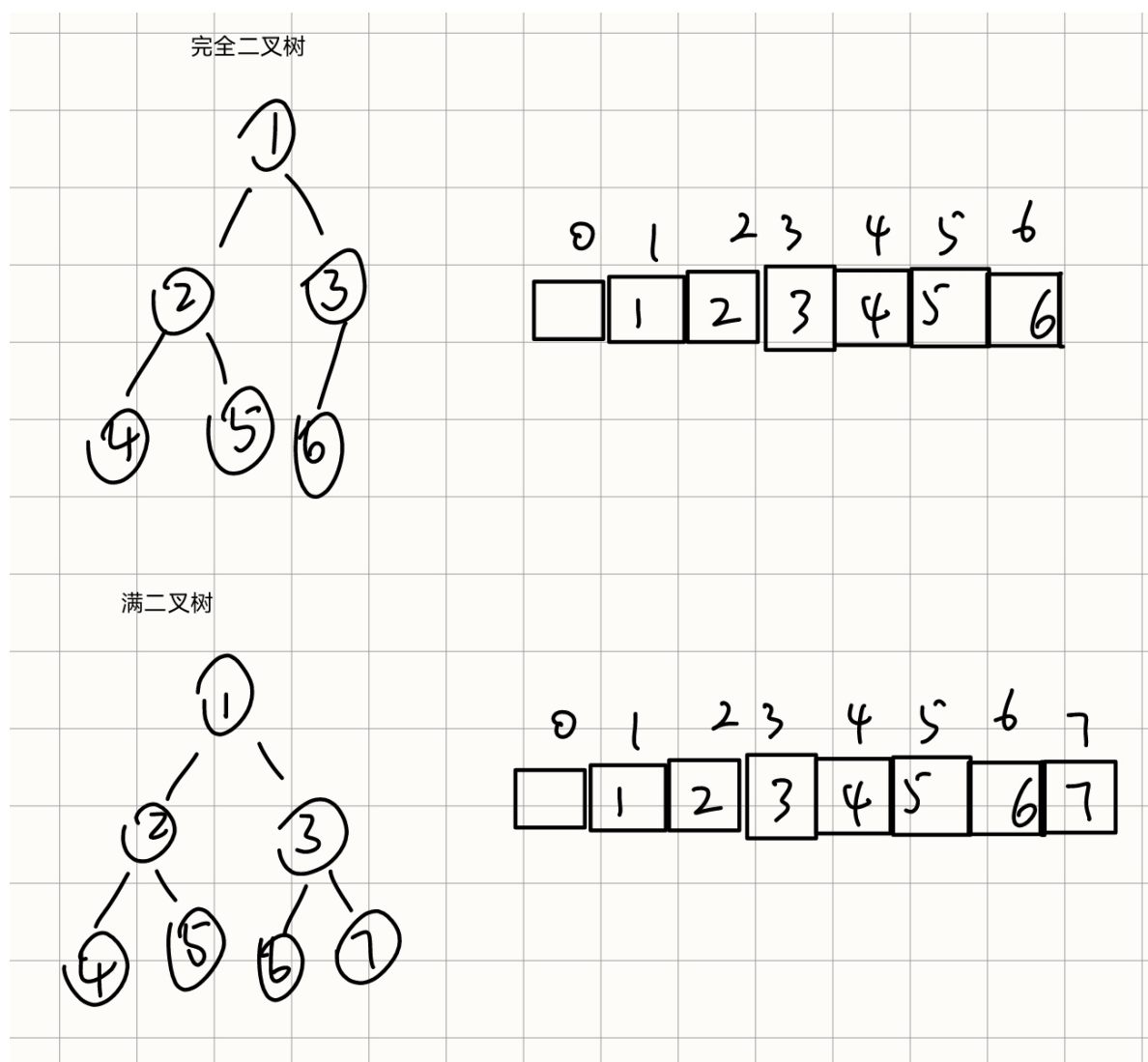
蓝蓝B站首页：[蓝蓝希望你上岸呀B站首页](#)

蓝蓝公众号：[应用题训练营专题](#)

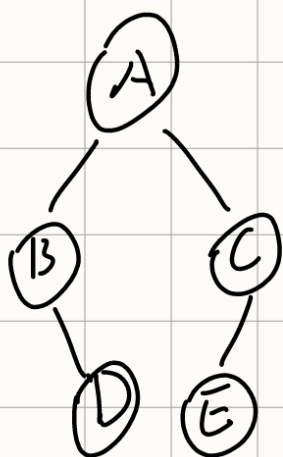
01、画完全二叉树，顺序存储完全二叉树，注意数组下标



02、画满二叉树，顺序存储满二叉树，注意数组下标



03、中序线索二叉树的构造

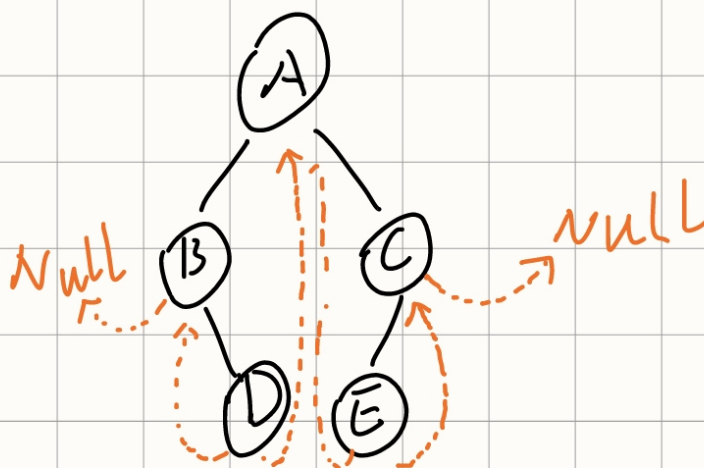


中序线索二叉树

对于每一个结点，若其左指针为空，就将它指向中序前驱，若其右指针为空就将它指向它的中序后继

中序遍历

B D A E C



Date / /

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

1. 完全二叉树 顺序存储

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

若 1 0 开始则 1 的左孩子为 1/2 为 1/2

2. 满二叉树 顺序存储

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

3. 中序遍历二叉树 LNR

BDAEC

Null

中序遍历中序遍历二叉树:

```

void CreateInThread (ThreadTree T) {
    ThreadTree pre = NULL; // 1. 第一个结点为序即 NULL
    if (T != NULL) {
        InThread (T, pre);
        pre->rchild = NULL;
        pre->rtag = 1; // 1. pre 为中序最后一个结点
    }
}

```

中序遍历二叉树 线索化中序遍历

```

void InThread (ThreadTree &p, ThreadTree &pre) {
    if (p != NULL) {
        InThread (p->lchild, pre); // 1. 左子树
        if (p->lchild == NULL) {
            p->lchild = pre;
            p->ltag = 1;
        }
        if (pre != NULL && rchild == NULL) {
            pre->rchild = p;
            pre->rtag = 1;
        }
        pre = p;
        InThread (p->rchild, pre); // 1. 右子树
    }
}

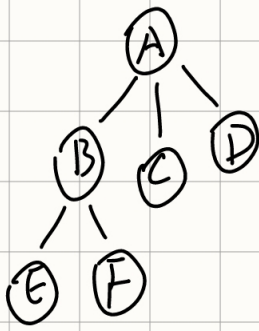
```

04、写出双亲表示法、孩子表示法、孩子兄弟表示法

双亲表示法

data Parent

0	A	-1
1	B	0
2	C	0
3	D	0
4	E	1
5	F	1

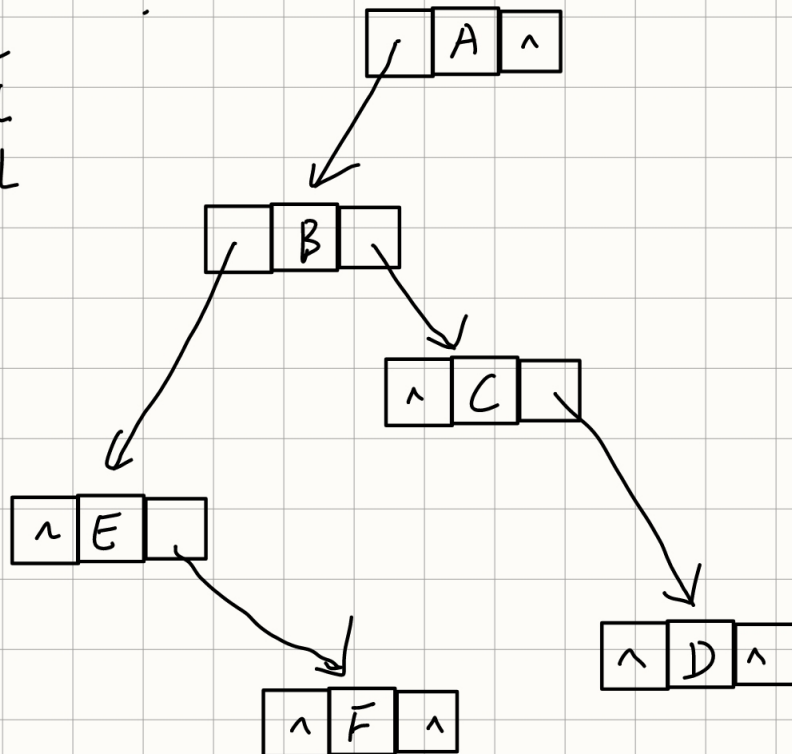


孩子表示法

data child

0	A	→ B → C → D
1	B	→ E → F
2	C	→ null
3	D	→ null
4	E	→ null
5	F	→ null

孩子兄弟表示法

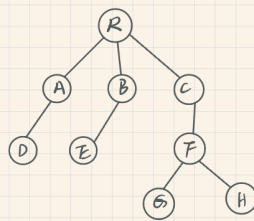




Date / /

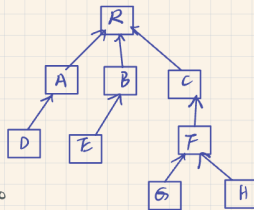


4 树与二叉树



① 树的双亲表示法

双亲指针图示



0	R	-1
1	A	0
2	B	0
3	C	0
4	D	1
5	E	2
6	F	3
7	G	6
8	H	6

双亲表示

#define MAX_TREE_SIZE 100

typedef struct {

int data;

int parent; // 双亲序号

} PTNode;

typedef struct {

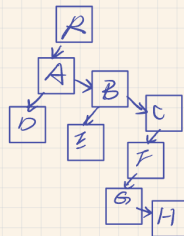
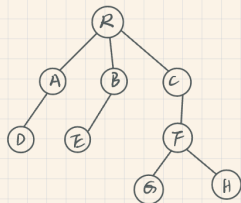
PTNode nodes[MAX_TREE_SIZE]; // 双亲表

int n;

// 结点数

} PTree;

② 孩子兄弟表示法



左指针指向第一个孩子,右指针指向下一个兄弟

表示代码如下:

typedef struct CSNode {

int data;

struct CSNode *firstChild, *nextSibling;

} CSNode, *CSTree;

孩子兄弟表示法与孩子表示法一样均存在找孩子容易找双亲难的问题



Date / /

