

$\leftarrow \rightarrow BCBBD$

ACDAA

DCDDL

ABABC

ABCDC

D

$\leftarrow \rightarrow$

1>

二.

(1) 链式

(2) 主要

(3)  $P \neq \text{NULL} \& \& P \rightarrow \text{next} = \text{NULL}$

(4)  $((), b, c)$

(5)  $top == 0$

(6)  $e = \text{SI}[top++]$

(7)  $e$

(8) ~~183~~

(9)  $i$

(10)  $P$

(11)  $n-1$

(12)  $pr \rightarrow \text{next} = p \rightarrow \text{next}$

(13)  $j < n$

(14)  $at[i]$

(15)  $2*i+2$

(16) 补码

~~18~~

(17)  $(x-1) \% 5 == 0 \& \& (x-5) \% 6 == 0 \& \& (x-4) \% 7 == 0 \& \& (x-1) \% 11 == 0$

(18)  $\text{int sum} = 0$

(19)  $i \% 3 == 0$

第 页 (共 页)

(20)  $i--$

(21) 8

(22)  $\text{static int arr[4]} = \{0\};$

(23)  ~~$*(\ast lp + 1) + 2 = 12$~~

(24)  $\ast str1 - \ast str2$

(25)  $str1++, str2++$

(26)  $break$

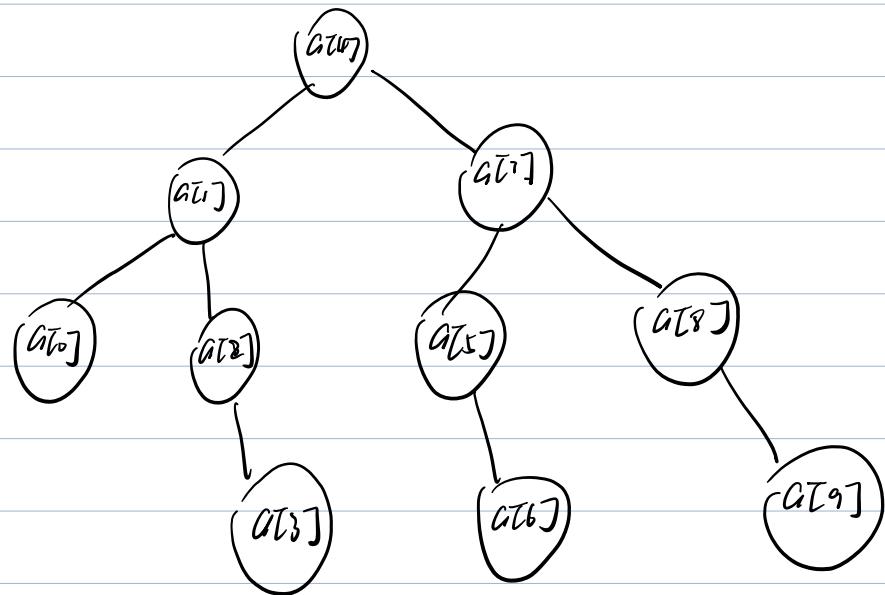
(27)  $(\text{NODE} *) \text{malloc}(\text{sizeof}(\text{NODE}))$

(28)  $head$

(29)  $P \neq \text{NULL}$

(30)  $P = P \rightarrow \text{next}$

三.



12) 2.9

13) 4

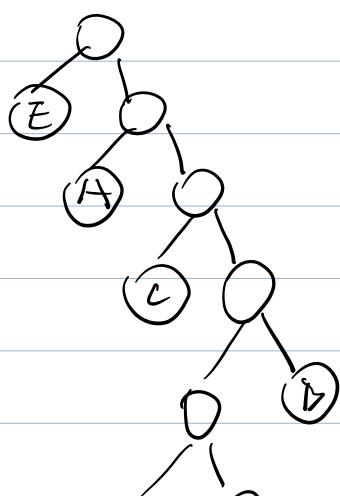
2.11) 1 2 0 5 3 4 6 8 7 9

12) 8 7 6 5 3 1 4 2 0 9

13) 1 2 0 4 3 8 6 5 9 7

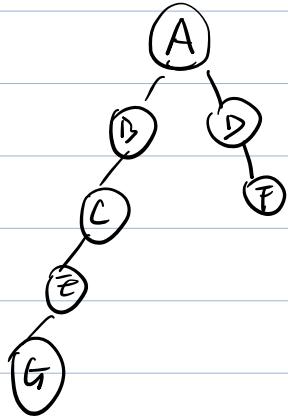
3. 11

12) 119



(7) (D)

4. ABCEGDF



121 A B C D E F G  
0 1 7 3 9 6 11  
0 5 8 3 10 6 11

最近时间

5. C C E E G G

6. 1, 3, 5, 6, 0,

7. 10

8. unknown

9. ABCADDB

10. 2, 5, 9, 14, 20.

IV (1)

```
int data;
struct node *lchild,*rchild;
}*BiTree,BiNode;
int func(BiTree root)
{
    if(root)
    {
        if(root->data<0)
            return 1+ func(root->rchild)+ func(root->lchild);
        else
            return func(root->rchild)+ func(root->lchild);
    }
    return 0;
}
```

12/12

```
#define size 10
typedef struct
{
    int data[size];
}Nums;

void Inputdata(Nums &nums)
{
    int i=0;
    while(i<size)
    {
        printf("请输入第%d个数\n",i)
        scanf("%d",&nums.data[i]);
        i++;
    }
}

void ComputeExtrem(Nums nums, int &min,int &minlocal,int &max,int &maxlocal)
{
    min=nums.data[0];max=nums.data[0];
    minlocal=0;maxlocal=0;
    for(int i=1;i<size;i++)
    {
        if(nums.data[i]<min)
{
{
            min= nums.data[i];
            minlocal=i;
}
else if(nums.data[i]> max)
{
            max= nums.data[i];
            maxlocal=i;
}
    }
}

int main()
{
    Nums test;
    int max,maxlocal,min,minlocal;
    Inputdata(test);
    ComputeExtrem(test, min, minlocal, max, maxlocal);
    return 0;
}
```