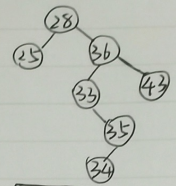
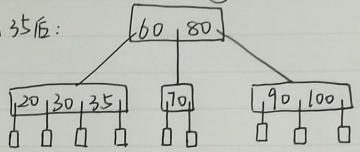
题二.3. 先序: 28, 25, 36, 33, 35, 34, 43.

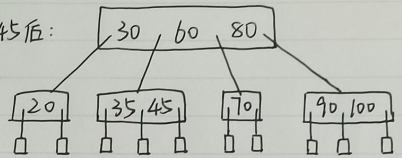
二叉搜索树为:



8. 插入35后:



插入45后:



11. 判定二叉树是否为AVL搜索树 if(左>根 || 根>右 || ||>1)

```
if ((t->lChild->element > t->element) || (t->rChild->element < t->element))
    flag = 0;
if(abs(Height(t->rChild) - Height(t->lChild)) >1)
    flag = 0;
Check (t->rChild);
Check (t->lChild);
```

13. $h \le 1+\log_{[2]}(N+1)/2$

$2 \le 1+\log_3(N+1)/2$

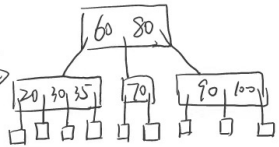
$\log_3(N+1) \ge 2$

$N+1 \ge 6$ $N \ge 5$ ∴最少有5个元素

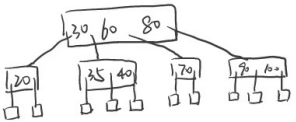
KOKUYO

8.

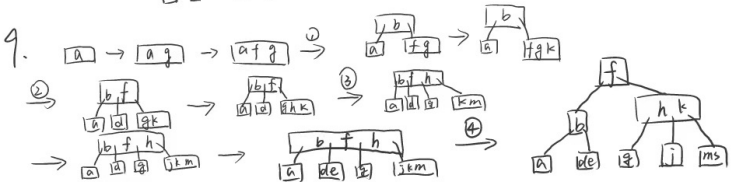
插入35



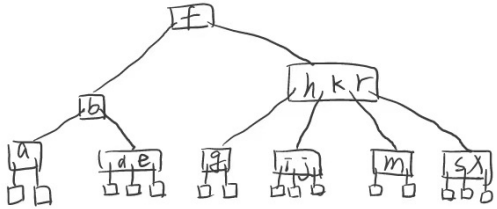
插入40



9.



⑤



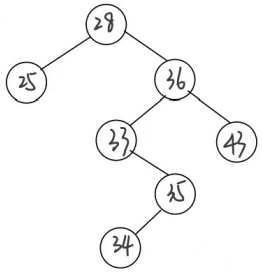
13. $2 \le 1 + \log_{\lceil \frac{3}{2} \rceil} (N+1)/2$

$1 \le \log_3 (N+1)/2$

$2 \le \log_3 (N+1)$ $\quad N+1 \ge 3^2 \quad \therefore N \ge 5$

树中元素个数最少为5

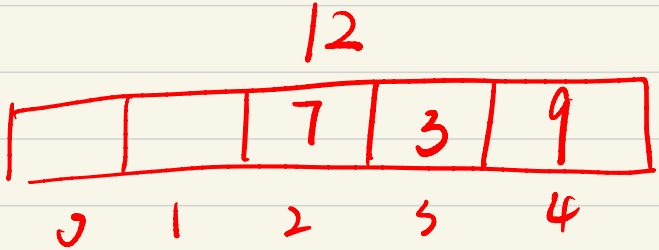
3.

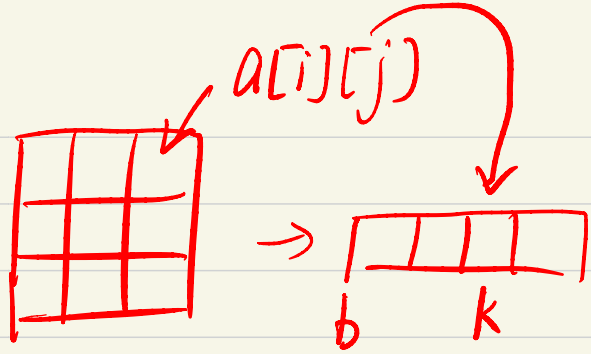


11.

```
if((t->rChild->element < t->element)||(t->lChild->element > t->element))
    if( abs(Hight(t->rChild)-Hight(t->lChild))>1 )
```

key 3 7 9 12





$loc = key \% 5$

$f(x) = nx + 1$

```
// 不带权的图的边搜索
Bool Search (MGraph *mg , int u, int v)
{   if (u<0 || v<0 || u>mg.n-1 || v>mg.n-1 || u==v)
        return 0;
    if ( mg.a[i][j] == 1)
        return 1;
    return 0;
{
```

// 网的边搜索

```
Bool Search (MGraph *mg , int u, int v)
{   if (u<0 || v<0 || u>mg.n-1 ||v>mg.n-1 || u==v)
        return 0;
    if ( mg.e[i][j] == mg.noEdge )
            return 0;
    return 1;
}
```

```
// 不带权图的V顶点的出度
int OutDegree(MGraph *mg, int v)
{
    if(v<0||v>mg.n-1) return 0;
    int cnt=0;    int i;
    for(i=0;i<mg.n;i++)
    {
        if(mg.a[v][i]==1) cnt++;
    }
    return cnt;
}
```

// 入度

```
int InDegree(MGraph *mg, int v)
{
    if(v<0||v>mg.n-1) return 0;
    int cnt=0;
    for(int i=0;i<mg.n;i++)
        if(mg.a[i][v]==1) cnt++;
    return cnt;
}
```