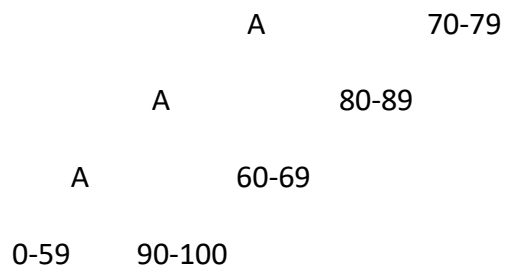


1. 一个分数转换程序，要把百分制成绩转为等级制，已知分数的分布如下，请问怎样分别用haffman树和最优二叉搜索树减少程序中的比较次数。

分数	比例	等级
0-59	0.05	E
60-69	0.15	D
70-79	0.40	C
80-89	0.30	B
90-100	0.10	A

-----考虑用haffman树，构造的结果如下：

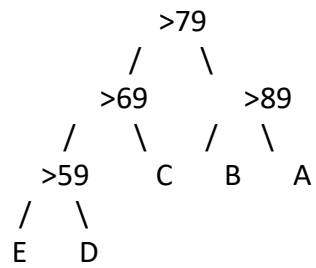


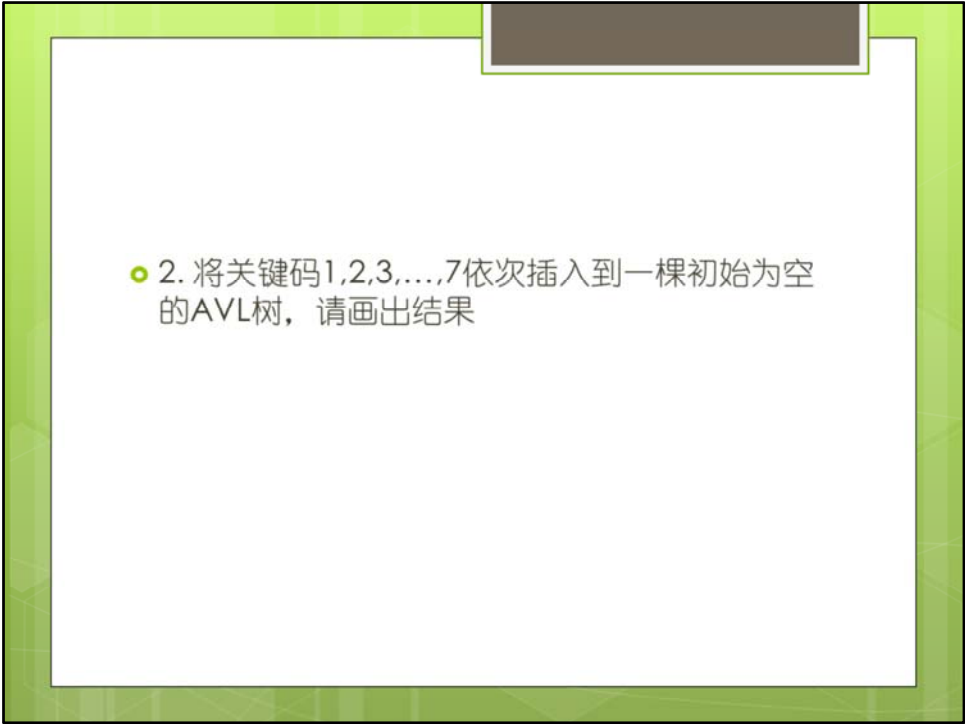
问题在于每个结点要判断2端，而且没有保持顺序。改造后得到结果应与根据最优二叉搜索树构造得到的一致。

-----考虑改为最优二叉搜索树  
则考虑几个判断点：59，69，79，89

分数	比例	
0-59	0.05	=q0+p1
60-69	0.15	=q1+p2
70-79	0.40	=q2+p3
80-89	0.30	=q3+p4
90-100	0.10	=q4+p5

需要进一步确定具体的 $q_i$ 和 $p_i$ 的值，因为每次要算 $w_{ij}=w_i(j-1)+p_j+q_j$ ，如果简单假设所有的 $p_i=0$ ，则  
 可以计算 $w_{01}=q_0+p_1+q_1, \dots$   
 然后计算各种 $c_{ij}$ ，...  
 最后得到的应该如下：



- 
- 2. 将关键码1,2,3,...,7依次插入到一棵初始为空的AVL树，请画出结果

答，1,2,3,...,2<sup>k</sup>-1依次插入后会形成一棵满二叉树。每次新结点总是在右下方插入，形成不平衡时总是做左单旋，当2<sup>k</sup>-1依次插入后左单旋完成后总是成为满二叉树。

- 3. Hash函数 $H(k)=3*k \% 11$ ，采用线性探测开放地址法处理冲突，试在0-10的散列空间中对关键字序列 (22, 41, 53, 46, 30, 13, 01, 67) 构造Hash表。并计算其查找成功的平均搜索长度。

答：

散列地址：	0	1	2	3	4	5	6	7	8	9
数据：	22		41	30	01	53	46	13	67	

$$ASL_{succ}=(1+1+1+1+2+2+2+6)/8=2$$