算法与数据结构

No.
Date

哈夫曼树（Huffman）

1. 目的：求带权外部路径长度最小的扩充二叉树，或称最优二叉树。

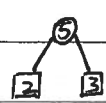
2. 基本思想：每次从所有扩充二叉树中，选取根结点权值最小和次最小的两棵，将它们作为左、右子树，成为新二叉树，其权值为左右根结点权值之和。直到只有一棵树为止。

3. 证明：不知道如何证明。

4. 算法

```
PHtTree huffman (int m, int *w)
{ pht = malloc (sizeof (struct HtTree));
  pht->ht = malloc ((2*m-1) * sizeof (HtTree));   (ht[2m-1]的数组)
①  for (i=0; i<2*m-1; i++){
       pht->ht[i].llink (rlink, parent) = -1.
       if (i<m) pht->ht[i].ww = w[i]; else pht->ht[i].ww=-1; }
   for (i=0; i<=m-1; i++){
       m1 (m2) = MAX; x1 (x2) = -1;
       for (j=0; j<m+i; j++)                    (找两个最小权无父结点的结点)
②          if (pht->ht[j].ww<m1 && pht->ht[j].parent=-1)
               {m2=m1; x2=x1; m1=pht->ht[j].ww; x1=j;}
           else if (pht->ht[j].ww<m2 && pht->ht[j].parent=-1)
               {m2 = pht->ht[j].ww; x2 = j;}.
       pht->ht[x1].parent = pht->ht[x2].parent = m+i;
       pht->ht[m+i].ww = m1+m2 (llink=x1, rlink=x2);
   }
   pht->root = 2*m-2; return pht;
}
```

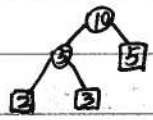
5. 图示：W[8] = {2, 3, 5, 11, 17, 23, 29, 31}.

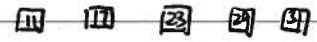
① 

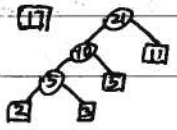
[5] [11] [17] [23] [29] [31]

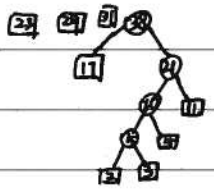
No.

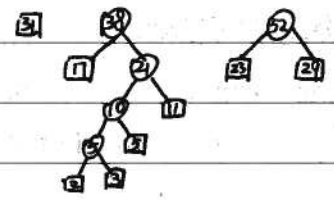
Date

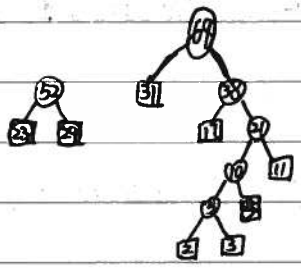
② 

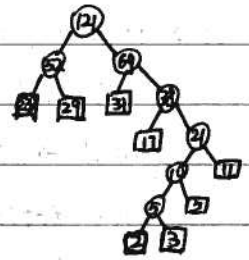


③ 

④ 

⑤ 

⑥ 

⑦ 

空 $O(m)$ 时 $O(m^2)$