



哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念

构造算法

k 叉哈夫曼树

例题

合并果子

哈夫曼编码

编码

解码

练习

哈夫曼树

Huffman Tree

河南省实验中学信息技术组

2026 年 02 月 23 日



知识回顾

哈夫曼树

河南省实验中学
信息技术组

- 堆

引例

哈夫曼树

概念

构造算法

k 叉哈夫曼树

例题

合并果子

哈夫曼编码

编码

解码

练习



【引例】成绩转换

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- 编写一个百分制转五级制的程序。

```
1 if(a < 60)
2   cout << " 不及格";
3 else if(a < 70)
4   cout << " 及格";
5 else if(a < 80)
6   cout << " 中等";
7 else if(a < 90)
8   cout << " 良好";
9 else
10  cout << " 优秀";
```

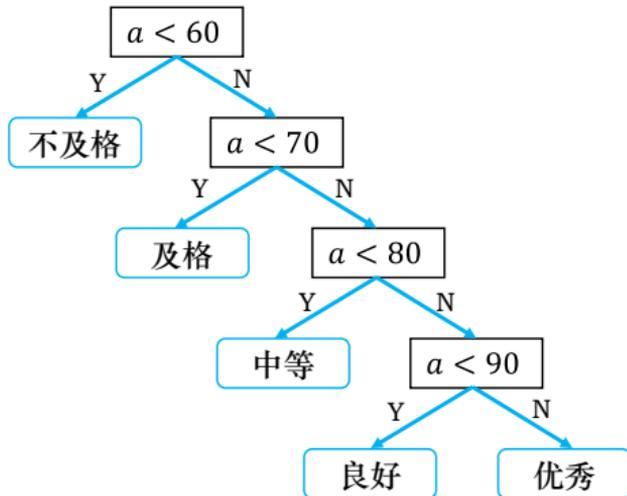


图: 判断树

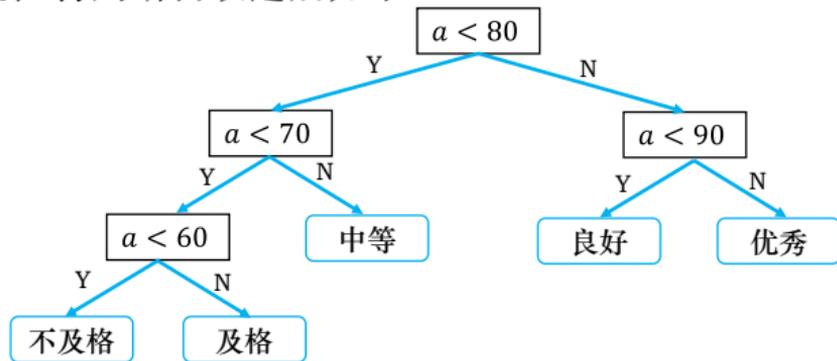


【引例】成绩转换

- 在实际中，如果有 100 个学生，学生成绩在 5 个等级上的分布规律如下表。

分数	0 ~ 59	60 ~ 69	70 ~ 79	80 ~ 89	90 ~ 100
人数	5	15	40	30	10

- 从上表可以看出 70 分以上的人数大约占总人数的 80%，而这些成绩都需要经过至少 3 次判断才能得到结果，当数据量很大时，必然浪费时间。
- 从表中可以看到，中等成绩人数最多，其次是良好成绩，那么我们让人数多的成绩先判定，将判断树改造后如下。



- 通过计算可知，效率提高了，效率提高的依据是什么？为什么这么设计效率能提高？



相关概念

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- 路径：从树中一个结点到另一个结点之间的分支构成两个结点间的路径。
- 路径长度：路径上分支的条数。树的路径长度是根结点到每一个结点的路径长度之和。
- 带权路径长度 (Weight Path Length, WPL)：树中每个结点都有权值，某结点到根结点路径长度与结点的权值的积称为带权路径长度。
- 树的带权路径长度为树中所有叶子结点的带权路径长度之和。



【引例】成绩转换

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

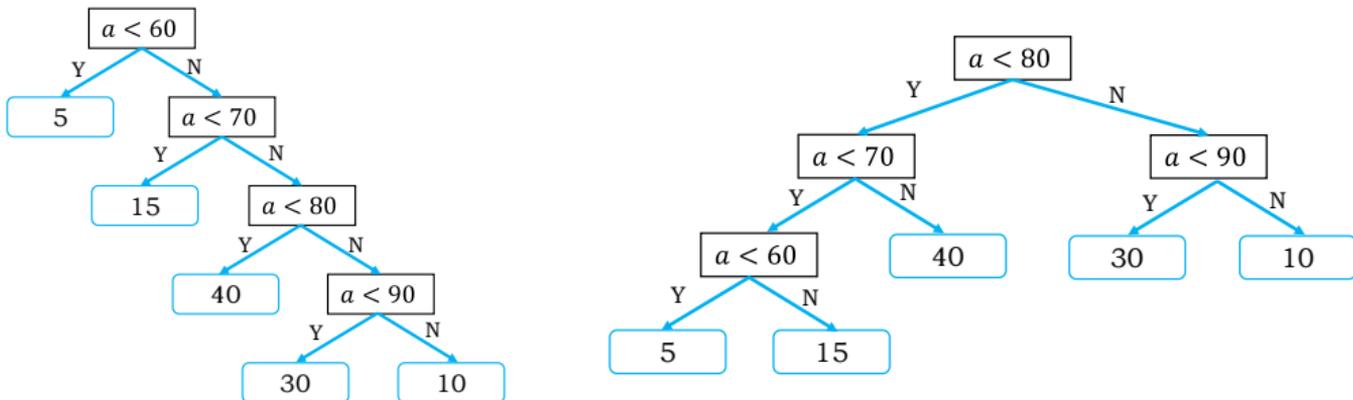
合并果子

哈夫曼编码

编码
解码

练习

- 现在我们将前面得到的两种判断树处理成带权树的形状，叶子结点内对应成绩的人数。



- $WPL_1 = 5 \times 1 + 15 \times 2 + 40 \times 3 + 30 \times 4 + 10 \times 4 = 315$ 。
- $WPL_2 = 5 \times 3 + 15 \times 3 + 40 \times 2 + 30 \times 2 + 10 \times 2 = 220$ 。
- 此处的带权路径长度表示总的判定次数，很明显第2种判断树更优，有没有更优的判断树呢？最优的判断树就是哈夫曼树。



哈夫曼树

假设有 n 个权值 $\{w_1, w_2, \dots, w_n\}$ ，构造一棵具有 n 个叶子结点的二叉树，每个叶子结点带权 w_i ，每个叶子结点到根结点路径长度 l_i ，则树的带权路径长度定义

$$\text{WPL} = \sum_{i=1}^n w_i * l_i$$

其中，带权路径长度最小的二叉树称为二叉哈夫曼树 (Huffman Tree)。

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念

构造算法

k 叉哈夫曼树

例题

合并果子

哈夫曼编码

编码

解码

练习



哈夫曼树构造算法

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- ① 首先，构造出 n 棵只有根结点的二叉树森林，其中第 i 棵二叉树的根结点权值为 w_i ；
- ② 选取两棵根结点权值最小的树作为左右子树构造一棵新的二叉树，且置新的二叉树的根结点的权值为其左右子树上根结点的权值之和；
- ③ 删除这两棵树，并将新得到的二叉树加入森林中。
- ④ 重复步骤 2 和步骤 3，直到森林中只包含一棵树为止，这棵树便是哈夫曼树。



哈夫曼树构造过程

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

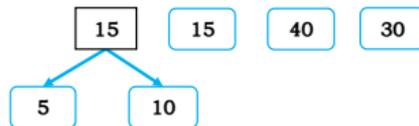
编码
解码

练习

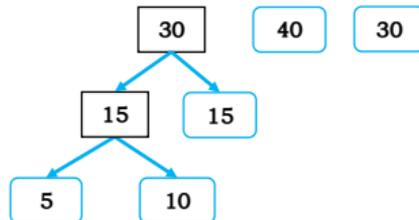
- 以百分制转五级制为例，构造 5 棵只有根结点的树。



- 在其中选择两棵根权值最小的树 5 和 10 为左右子树构造新的二叉树，新二叉树的根结点的权值为两棵子树权值的和 15。删除这两棵子树，将新的二叉树加入森林。



- 在其中选择两棵根权值最小的树 15 和 15 为左右子树构造新的二叉树，新二叉树的根结点的权值为两棵子树权值的和 30。删除这两棵子树，将新的二叉树加入森林。





哈夫曼树构造过程

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k 叉哈夫曼树

例题

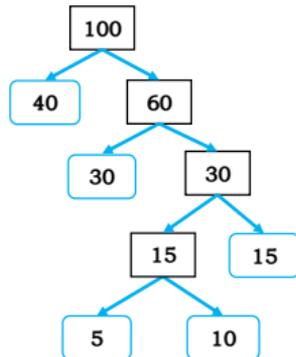
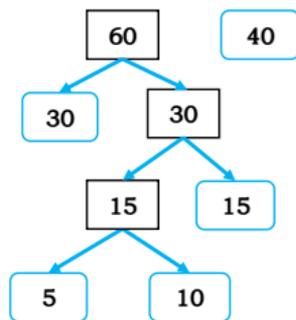
合并果子

哈夫曼编码

编码
解码

练习

- 在其中选择两棵根权值最小的树 30 和 30 为左右子树构造新的二叉树，新二叉树的根结点的权值为两棵子树权值的和 60。删除这两棵子树，将新的二叉树加入森林。
- 在其中选择两棵根权值最小的树 40 和 60 为左右子树构造新的二叉树，新二叉树的根结点的权值为两棵子树权值的和 100。删除这两棵子树，将新的二叉树加入森林。
- 集合中只有一棵树，算法结束。
- $WPL_3 = 40 \times 1 + 30 \times 2 + 15 \times 3 + 5 \times 4 + 10 \times 4 = 205$ 。
- 在实际应用中，因为每次判断都需要两次比较(例如根判断写为 $a \geq 70 \& \& a < 80$)，所以性能反而不如判断树 2。





哈夫曼树构造算法

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- 具体实现时，可以用最小堆来存储已经建成的子树(根)，堆中元素按照子树根结点权值来进行调整。
- 因此，取出两棵根结点权值最小的树相当于进行两次出堆操作，然后利用出堆的两棵子树建立一棵新的子树，然后将新建成的子树(根)入堆。
- 最后，堆中只有一棵树，这棵树就是哈夫曼树。
- 时间复杂度： $O(N \log N)$ 。
- 在信息学竞赛中，一般不会考察建立具体的哈夫曼树，而是直接求出哈夫曼树的带权路径长度。其中，最经典的例题是 NOIP 2004 中的合并果子。



k 叉哈夫曼树

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
 k 叉哈夫曼树

例题

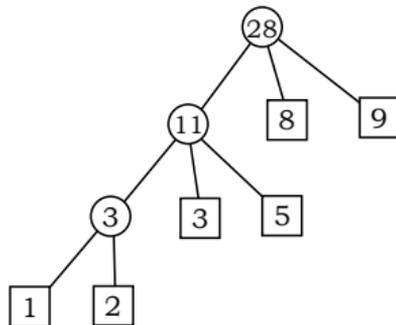
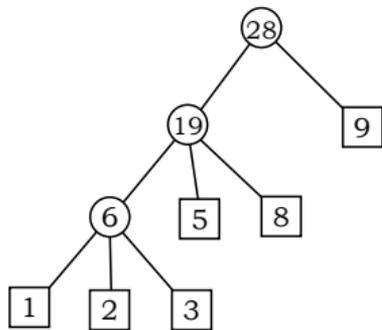
合并果子

哈夫曼编码

编码
解码

练习

- 对于 k 叉哈夫曼树，最直观的方法是每次从堆中取出最小的 k 个权值构造树，这种方法可行吗？
- 仔细思考会发现，如果在最后一次构造树时，堆的大小（堆中树的个数）在 $2 \sim k-1$ 之间，那么整个哈夫曼树的根的子结点数就少于 k ，这显然不是最优解，因为此时如果任取一个深度稍大的结点，将它改为根结点的子结点，一定会更优。
- 例如，左图的带权路径之和为 53，右图的带权路径之和为 42。





k 叉哈夫曼树

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
 k 叉哈夫曼树

例题

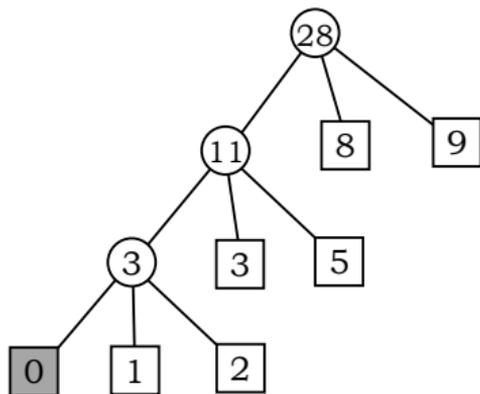
合并果子

哈夫曼编码

编码
解码

练习

- 在执行建树之前，可以补加一些权值为 0 的叶子结点，使得叶子结点的个数满足 $(n - 1) \bmod (k - 1) = 0^1$ ，然后执行“每次从堆中取最小的 k 个权值”的操作建树就是正确的。



¹让所有的内部结点都有 k 个孩子，设叶子结点有 n 个，内部结点有 m 个，则 $km = m + n - 1 \Rightarrow n - 1 = m(k - 1) \Rightarrow (n - 1) \bmod (k - 1) = 0$ 。



【例】合并果子

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k 叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

在一个果园里，多多已经将所有果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 $n - 1$ 次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1,2,9。可以先将 1,2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为 12。所以多多总共耗费体力为 $3+12=15$ 。可以证明 15 为最小的体力耗费值。



【例】合并果子

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

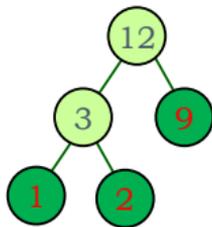
合并果子

哈夫曼编码

编码
解码

练习

- 每次合并消耗的体力等于两堆果子的重量之和，所以最终消耗的体力总和就是每堆果子的重量乘上它参与合并的次数。
- 这恰好是一个哈夫曼树，果子的重量就是叶子结点的权值，参与合并的次数就是叶子结点到根结点的距离。



- 利用果子的重量建立一个最小堆，不断取出堆中最小的两个值，将这两个值累加到答案(消耗的体力)，同时将它们的和插入堆中。
- 当堆的大小为 1 时，算法结束，输出累加的答案即可。



【例】合并果子

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

```
1 for(int i = 1; i <= n; ++i) pq.push(a[i]);
2 int ans = 0;
3 while(pq.size() > 1)
4 {
5     int b = pq.top(); pq.pop();
6     int c = pq.top(); pq.pop();
7     ans += b + c;
8     pq.push(b + c);
9 }
```



哈夫曼编码

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- 哈夫曼树最经典的应用是在通信领域，目的是为了**解决远距离通信的传输效率问题**。
- 假设我们有一段文字“BADCADFEED”需要传输给别人，显然我们需要用二进制进行传输。这段文字中只出现了 ABCDEF 六个字母，则我们可以造出如下编码表

字母	A	B	C	D	E	F
编码	000	001	010	011	100	101

根据上表，传输序列为“001000011010000011101100100011”，长度为 30，对方接收后 3 位一分来解码即可。

- 这种编码方式被称为**定长编码**。



哈夫曼编码

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- 但是在实际应用中各个字母出现的频率不同，例如‘A’和‘X’出现的频率前者高后者低，那么利用定长编码方式会造成冗余。因此，我们需要设计一种变长编码，希望出现频率高的字符的编码短，出现频率低的字符编码长，这样可以提高传输效率。
- 我们重新设计一个编码表如下

字母	A	B	C	D	E	F
编码	111	010	011	10	00	110

- 文字“BADCADFEED”的传输序列为“0101111001111110110000010”，长度为25，比定长编码节约了16.7%的传输成本，随着传输数据量的增加，这种节约更大。
- 变长编码问题：
 - 长短不一的编码很容易出现歧义，所以在设计过程中必须保证任一字符的编码都不是另一个字符的编码的前缀，这么编码也称为前缀编码。
 - 如何设计这种编码？对方如何解码？



哈夫曼编码

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- 为了衡量各种变长编码的效率，定义总编码长度

$$\sum_{i=1}^n f_i l_i$$

其中 f_i 为第 i 个字符出现的次数， l_i 为第 i 个字符的编码长度。编码长度最小的前缀编码被称为最优编码。

- 利用各字符出现的次数为叶子结点构造一棵哈夫曼树，规定哈夫曼树的左分支代表 0，右分支代表 1，则从根结点到叶子结点所经过的分支组成的 01 序列便是该叶子结点对应字符的编码，这就是哈夫曼编码。
- 信息学奥赛中一般不会考察建立具体的哈夫曼编码树，而是求哈夫曼编码的总编码长度，即哈夫曼树的带权路径长度。



编码

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

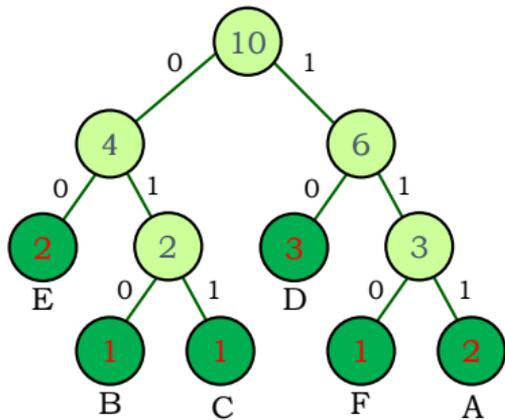
哈夫曼编码

编码
解码

练习

- 右图为“BADCADFEED”根据字符出现次数生成的哈夫曼树。
- 编码：对右边的树进行遍历，遍历过程中记录走左分支(0)还是右分支(1)，到叶子结点后即可得到叶子结点的对应字符的编码。

字符	次数	编码
A	2	111
B	1	010
C	1	011
D	3	10
E	2	00
F	1	110



图：哈夫曼编码树



编码

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

```
1 void preorder(int x, string s)
2 {
3     if(!lc[x] && !rc[x]) { cout << c[x] << ":" << s << "\n"; return ;}
4     if(lc[x]) preorder(lc[x], s + "0");
5     if(rc[x]) preorder(rc[x], s + "1");
6 }
7
8 for(int i = 0; i < s.size(); ++i) ++f[s[i] - 'A']; // 字符出现次数
9 // 权值和结点编号
10 priority_queue<pair<int, int>, vector<pair<int, int>>,
11     greater<pair<int, int>>> pq;
12 for(int i = 0; i < 26; ++i) // 叶子结点
13     if(f[i]) ++n, c[n] = 'A' + i, d[n] = f[i], pq.push({d[n], n});
14 while(pq.size() > 1) // 建立哈夫曼编码树
15 {
16     pair<int, int> x = pq.top(); pq.pop(); // 左孩子
17     pair<int, int> y = pq.top(); pq.pop(); // 右孩子
18     ++n, d[n] = x.first + y.first;
19     lc[n] = x.second, rc[n] = y.second;
20     pq.push({d[n], n});
21 }
22 rt = pq.top().second; // 根结点
23 preorder(rt, ""); // 获取每个字符的编码
```



解码

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

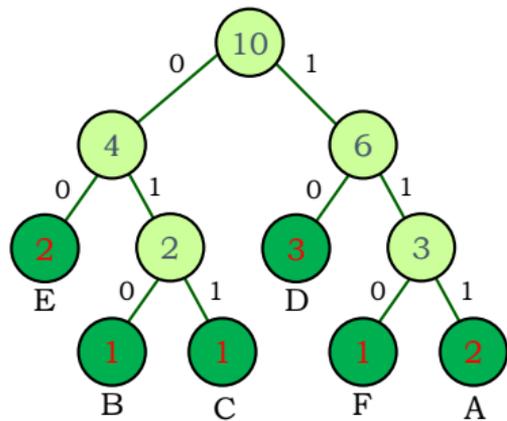
哈夫曼编码

编码
解码

练习

- 解码：对于收到的序列，从根结点开始，如果是0，向左走，如果是1，向右走，直到走到叶子结点即可得到字符，同时回到根结点继续下一个字符解码。

01011110011110110000010
B A D C A D F E E D



图：哈夫曼编码树



解码

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

```
1 void decode(string s)
2 {
3     int x = rt;
4     for(int i = 0; i < s.size(); ++i)
5     {
6         if(s[i] == '0') x = lc[x];    // 编码为 0 往左孩子去
7         else if(s[i] == '1') x = rc[x]; // 编码为 1 往右孩子去
8         if(!lc[x] && !rc[x]) cout << c[x], x = rt; // 到叶子结点 解码出一个字符
9     }
10 }
```



练习

哈夫曼树

河南省实验中学
信息技术组

引例

哈夫曼树

概念
构造算法
k叉哈夫曼树

例题

合并果子

哈夫曼编码

编码
解码

练习

- 合并果子 [NOIP 2004](COGS 75)
- 哈夫曼编码的长度 (COGS 3192)
- 哈夫曼编码 (COGS 3298)
- 荷马史诗 [NOI 2015](COGS 2021)