CSP 第一轮经典计算题型总结

逻辑表达式

- **题目**
  2020 CSP-J 第3题
  2013 NOIP普及组 第4题
  2010 NOIP普及组 第3题
- **经验**
  1. 注意基本概念即可，实际做题中可以考虑**代入计算**:
     (a) 与（数学 $\wedge$，信息学 `&&`）：同真为真。
     (b) 或（数学 $\vee$，信息学 `||`）：有真为真。
     (c) 非（数学 $\neg$，信息学 `!`）：真假颠倒。
  2. 优先级：非 $>$ 与 $>$ 或。

进制

- **题目**
  2022 CSP-J 第13题
  2021 CSP-J 第7题
  2018 NOIP普及组 第2题
  2017 NOIP普及组 第15题
  2016 NOIP普及组 第8题
  2016 NOIP普及组 第7题
  2016 NOIP普及组 第2题
  2014 NOIP普及组 第11题
  2013 NOIP普及组 第6题
  2013 NOIP普及组 第2题
  2012 NOIP普及组 第4题
  2011 NOIP普及组 第7题
  2011 NOIP普及组 第1题
- **经验**
  1. 有如下经典题型:
     (a) 不同进制数字间判断相等。
     (b) 不同进制数字间的计算。
     (c) 进制转换。
     (d) 小数进制转换。
  2. 二/八/十六进制之间可快速转换（二进制作为桥梁）：

$$(765)_8 = (111-110-101)_2 = (1-1111-0101)_2 = (1F5)_{16}$$

  3. 小数的进制转换:
     (a) 小数无论是什么进制，其整数部分就是整数部分，小数部分就是小数部分，所以可以分开进行转

换。

(b) k 进制小数转十进制：计算 $(0.31)_4 = (08125)_{10}$

$$(0.31)_4 = 3 \times 0.25 + 1 \times 0.0625 = (08125)_{10}$$

(c) 十进制小数转 k 进制（乘 k 取整法）：计算 $(0.8125)_{10} = (0.1101)_2$

步骤	计算	结果
1	$0.8125 \times 2 = 1.625$	小数点后第 1 位取 1
2	$0.625 \times 2 = 1.25$	小数点后第 2 位取 1
3	$0.25 \times 2 = 0.5$	小数点后第 3 位取 0
4	$0.5 \times 2 = 1.0$	小数点后第 4 位取 1

存储空间计算

- **题目**
  2019 CSP-J 第3题
  2018 NOIP普及组 第3题
  2017 NOIP普及组 第4题
  2015 NOIP普及组 第1题
  2015 NOIP普及组 第6题
  2015 NOIP普及组 第7题
  2014 NOIP普及组 第2题
  2013 NOIP普及组 第1题
  2012 NOIP普及组 第16题
  2011 NOIP普及组 第3题
- **经验**
  1. 一般计算的点就三个：
     (a) 计算存储空间容量：很多时候会直接给出。
     (b) 计算单个元素的占用容量：一般就是看需要多少 bit 来存储信息，例如：
        - 如果存储一个黑白像素点，那么用 1 个 bit 就可以了，大小 $1b$；
        - 如果存储 320×640 的黑白像素图像，那么用 320×640 个 bit，大小 $25KB$；
        - 如果存储 320×640 的彩色像素图像（共 54 种颜色），那么每个点至少需要 6 个 bit 来存储颜色信息，大小 $150KB$；
        - 如果是一个汉字字符，没说的话默认两个字节 16 个 bit，大小 $2B$。

     (c) 计算元素的存储个数上限，这个就是简单除法，不过要注意单位：

$$1TB = 2^{10}GB = 2^{20}MB = 2^{30}KB = 2^{40}B = 8 \times 10^{40}b$$

原码/反码/补码

- **题目**
  2017 NOIP普及组 第1题

2010 NOIP普及组 第11题

- **经验**
  1. 无非就三个概念两个结论：
     (a) 正数符号位为 0，负数符号位为 1。
     (b) 原码转反码：非符号位取反。
     (c) 反码转补码：加 1。
     (d) 正数三码一致。
     (e) 0 有两种原码、反码，一种补码。

二分查找

- **题目**
  2019 CSP-J 第5题
  2014 NOIP普及组 第18题
- **经验**
  1. 了解二分查找的原理以及操作方式即可。
  2. 二分查找的时间复杂度为 $log(n)$。

排序算法

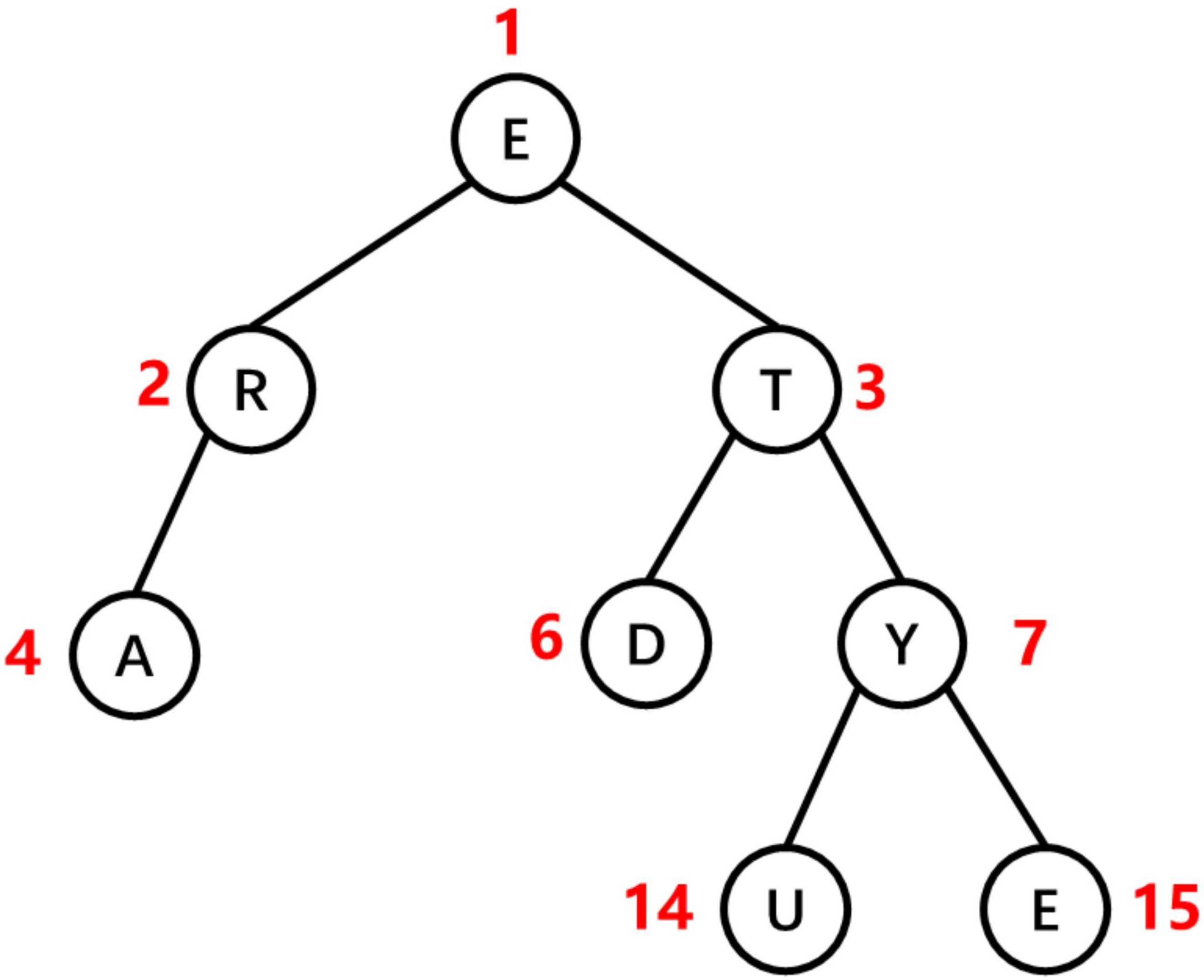
- **题目**
  2022 CSP-J 第12题
  2018 NOIP普及组 第8题
  2013 NOIP普及组 第14题
  2012 NOIP普及组 第8题
  2011 NOIP普及组 第8题
  2010 NOIP普及组 第12题
- **经验**
  1. 一般按各种排序的方法模拟即可，个别情况需要确定使用何种排序：
     (a) 只能相邻交换，那么使用冒泡排序的交换次数最少。
     (b) 可以任意交换，那么使用选择排序的交换次数最少。
  2. **稳定排序**：冒泡排序、插入排序、归并排序、计数排序、桶排序、基数排序；**不稳定排序**：选择排序、希尔排序、快速排序、堆排序。
  3. 注意快排没有优化时时间复杂度为 $O(n^2)$，优化后才是 $O(nlogn)$。

二叉树节点计算与存储

- **题目**
  2022 CSP-J 第8题
  2019 CSP-J 第8题
  2018 NOIP普及组 第7题
  2016 NOIP普及组 第11题
  2016 NOIP普及组 第二大题 第2题
  2015 NOIP普及组 第17题
  2014 NOIP普及组 第16题

2013 NOIP普及组 第9题
2011 NOIP普及组 第7题

- **经验**
  1. 熟悉基本概念，然后记得如下性质：
     (a) 在二叉树的第 i 层上最多有 2^{i-1} 个结点。
     (b) 深度为 k 的二叉树最多有 $2^k - 1$ 个结点。
     (c) 具有 n 个结点的完全二叉树的深度为 $\lfloor log_2^b \rfloor + 1$ 。
     (d) 结点数量等于不同度结点数量和：$n = num_0 + num_1 + num_2$ 。
     (e) 结点数量等于所有儿子数量和加一：$n = num_1 + 2 \times num_2 + 1$。
     (f) 上面两式可以推出：$n_0 = n_2 + 1$。
  2. 上述结论可以扩展到 k −叉树。
  3. 二叉树的存储（堆存储）使用顺序结构实现（比如数组）——根节点存储在 1 号位置，然后对于存储在 x 号位置的结点，其左儿子存储在 $2 \times x$ 号位置，右儿子存储在 $2 \times x + 1$ 号位置：



位置	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
值	E	R	T	A	-	D	Y	-	-	-	-	-	-	U	E

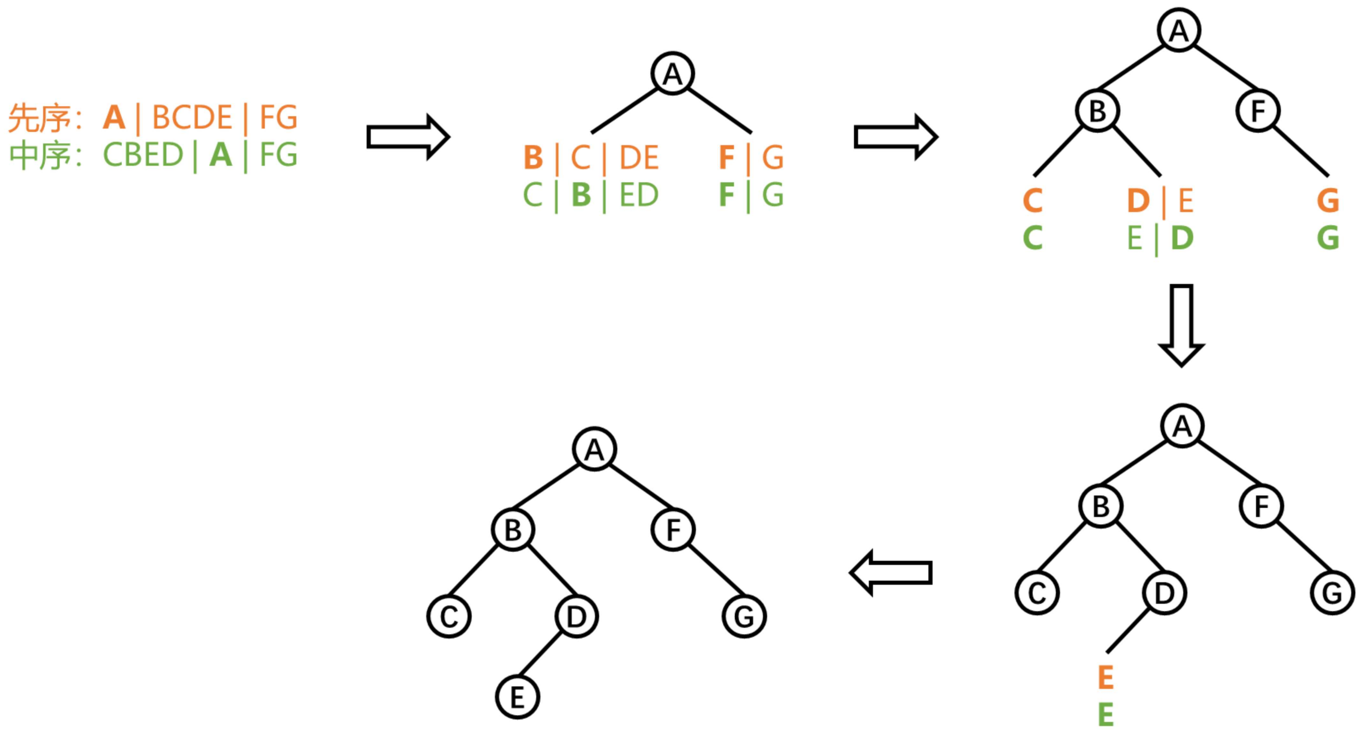
二叉树遍历

- **题目**
  2019 CSP-J 第14题
  2015 NOIP普及组 第16题

2013 NOIP普及组 第11题

2012 NOIP普及组 第6题

- **经验**：
  1. 前/中/后序遍历概念：
     (a) 前序遍历：根左右，根最前。
     (b) 中序遍历：左根右，根分割。
     (c) 后序遍历：左右根，根最后。
  2. 确定树结构：先/后序找根，中序用根分割



9.表达式前中后缀

- **题目**

  2022 CSP-J 第6题

  2021 CSP-J 第9题

  2017 NOIP普及组 第12题
- **经验**
  1. 前缀后缀转中缀一般是使用栈来操作：

     (a) 前缀（波兰式）转中缀：用栈存子表达式与运算符，如果遇到数字，且栈顶为子表达式，那么取出该子表达式与其前面的运算符，拼接后存储。举个例子：

栈	输入	剩余输入
空	`-`	`* + 3 4 5 6`
`-`	`*`	`+ 3 4 5 6`
`-,*`	`+`	`3 4 5 6`
`-,*,+`	`3`	`4 5 6`
`-,*,+,3`	`4`	`5 6`
`-,*,3 + 4`	`5`	`6`

`-,(3 + 4) * 5`	`6`	无
`(3 + 4) * 5 - 6`	无	无

（b）后缀（逆波兰式）转中缀：用栈存子表达式，如果遇到运算符，那么取出两个表达式，与新运算符拼接后存储。举个例子：

栈	输入	剩余输入
无	`24`	`8 + 3 * 4 10 7 - * /`
`24`	`8`	`3 * 4 10 7 - * /`
`24,8`	`+`	`3 * 4 10 7 - * /`
`24 + 8`	`3`	`4 10 7 - * /`
`24 + 8,3`	`*`	`4 10 7 - * /`
`(24 + 8) * 3`	`4`	`10 7 - * /`
`(24 + 8) * 3,4`	`10`	`7 - * /`
`(24 + 8) * 3,4,10`	`7`	`- * /`
`(24 + 8) * 3,4,10,7`	`-`	`* /`
`(24 + 8) * 3,4,10 - 7`	`*`	`/`
`(24 + 8) * 3,4 * (10 - 7)`	`/`	无
`((24 + 8) * 3) / (4 * (10 - 7))`	无	无

2. 中缀转前缀后缀：先补全括号，然后移动括号内运算符——将运算符移动到最前面是前缀，移动到最后面是后缀，移动完后把所有括号消除。举个例子：
    - 中缀转前缀：`(24 + 8) * 3 / (4 * (10 - 7))` → `(((24 + 8) * 3) / (4 * (10 - 7)))` → `(/ (* (+ 24 8) 3) (* 4 (- 10 7)))` → `/ * + 24 8 3 * 4 - 10 7`。
    - 中缀转后缀：`(24 + 8) * 3 / (4 * (10 - 7))` → `(((24 + 8) * 3) / (4 * (10 - 7)))` → `(((24 8 +) 3 *) (4 (10 7 -) *)) /` → `24 8 + 3 * 4 10 7 - * /`。

10.图理论

- **题目**
  2022 CSP-J 第9题
  2021 CSP-J 第8题
  2021 CSP-J 第6题
  2018 NOIP普及组 第11题
  2017 NOIP普及组 第10 题
  2016 NOIP普及组 第15题
  2015 NOIP普及组 第12题
  2014 NOIP普及组 第17题

2014 NOIP普及组 第二大题 第2题
2013 NOIP普及组 第10题
2013 NOIP普及组 第12题
2011 NOIP普及组 第5题
2011 NOIP普及组 第19题

- **经验**
  1. 图的各种基本概念要熟记。
  2. 基础性质类：
     - 当一个 n 个点的连通图拥有 $n-1$ 条边时，其是一棵树。
     - 当一个 n 个点的简单图拥有 m 条边时，其是一个包含 $n-m$ 棵树的森林。
     - 有向图中，所有结点的出度和等于所有结点的入读和等于边数量。
     - 有向图中，一个结点的出度与入度之和应小于等于边数量。
     - n 个点的无向完全图有 C_n^2 条边，n 个点的有向完全图有 A_n^2 条边。
     - 如果图是连通的且只有 2 个奇点，那么该图存在欧拉路径。
     - 如果图是连通的且没有奇点，那么该图存在欧拉回路。
  3. 图论算法：记住基本步骤即可，一般放选择里不会考太难手动模拟可以解决。
     (a) 最短路。
     (b) 最小生成树。
     (c) 拓扑序。
     (d) dfs 序、bfs 序。

组合数学+概率分析

- **题目**
  2021 CSP-J 第10题
  2020 CSP-J 第14题
  2020 CSP-J 第15题
  2019 CSP-J 第7题
  2019 CSP-J 第12题
  2019 CSP-J 第13题
  2018 NOIP普及组 第12题
  2018 NOIP普及组 第13题
  2017 NOIP普及组 第9题
  2017 NOIP普及组 第19题
  2016 NOIP普及组 第16题
  2016 NOIP普及组 第二大题 第1题
  2015 NOIP普及组 第二大题 第1题
  2014 NOIP普及组 第二大题 第1题
  2013 NOIP普及组 第二大题 第1题
  2012 NOIP普及组 第二大题 第1题
  2012 NOIP普及组 第二大题 第2题
  2011 NOIP普及组 第二大题 第1题
- **经验**
  1. 容斥原理：其扩展原理也稍微记忆一下。
  2. 排列与原排列：
     (a) 排列：$A_n^m = \frac{n!}{(m-1)!}$。

(b) 圆排列：$\frac{A_n^m}{m} = \frac{n!}{m!}$。

3. 组合：
   (a) $C_n^m = \frac{n!}{m!(n-m)!}$。
   (b) $C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$。
   (c) $C_n^m = C_n^{n-m}$。
   (d) $C_n^0 + C_n^1 + \ldots + C_n^n = 2^n$。
4. 错排：
   (a) $f_n = n! \times (\frac{1}{2!} + (-\frac{1}{3!}) + \frac{1}{4!} + \ldots + \frac{(-1)^n}{n!})$。
   (b) $f_n = (n-1)(f_{n-1} + f_{n-2}), f_1 = 0, f_2 = 1$。
5. 分配问题的三种相关解题技巧：
   (a) 捆绑法。
   (b) 插空法。
   (c) 分情况讨论，获取递归公式。
6. 概率问题经典技巧：
   (a) 分情况讨论。
   (b) 求补集概率，间接计算。

哈夫曼编码

- **题目**
  2022 CSP-J 第7题
  2021 CSP-J 第11题
  2011 NOIP普及组 第15题
- **经验**
  1. 哈夫曼是一种编码方式，不同于定长编码，哈夫曼根据字的出现频率调整编码长度（频率高的短，频率低的长）以达到压缩消息的效果。举个例子：

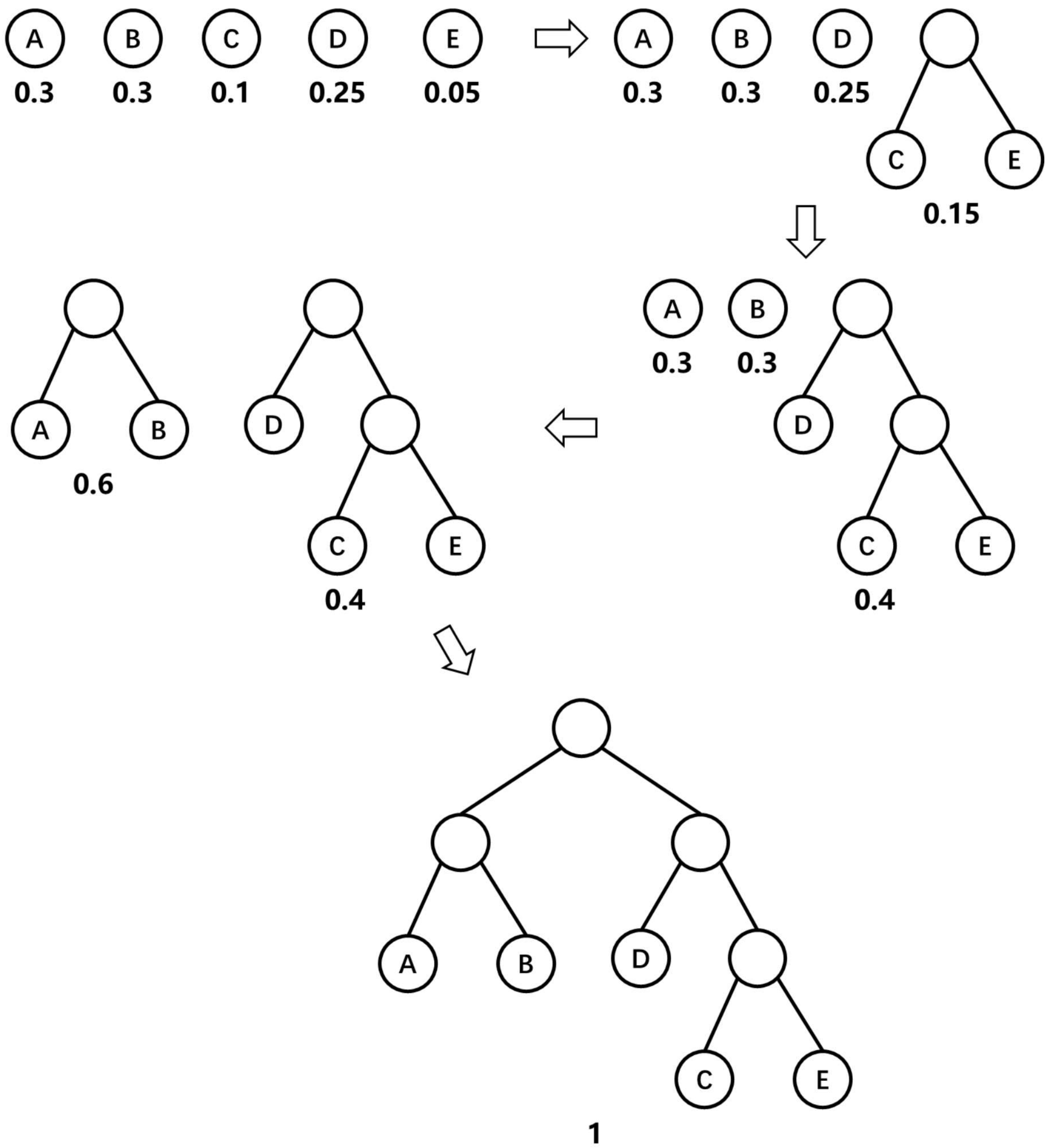
字	频率	传统编码	哈夫曼编码
A	0.5	`00`	`0`
B	0.3	`01`	`10`
C	0.15	`10`	`110`
D	0.05	`11`	`111`

此时有：
(a) 传统编码平均长度期望： 2。
(b) 哈夫曼编码平均长度期望：

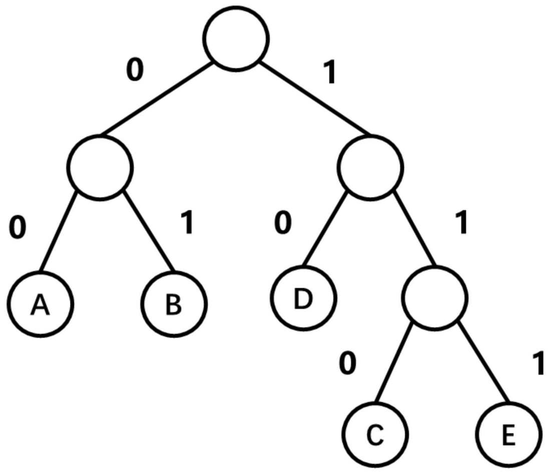
$$1 \times 0.5 + 2 \times 0.3 + 3 \times 0.15 + 3 \times 0.05 = 1.7$$

2. 哈夫曼编码的构造方式：
   (a) 将所有字变成一个单节点的树（权值为它们的频率）：

（b）每次选取两个权值最小的树（多个可任意），然后创造一个根节点连接它们（新树的权值为两棵子树权值之和），直到剩下一棵树：



（c）从根节点出发，左 0 右 1，叶子节点（字）的编码就是路径值：



字	哈夫曼编码
A	`00`

B	`01`
C	`111`
D	`10`
E	`110`

3.哈夫曼编码是异字头的，不会出现二义问题。
4.不仅仅是频率，频次等都可以作为权值。

位运算

- **题目**
  2019 CSP-J 第2题
- **经验**
  1. 记住每种位运算的概念和符号：
     (a) 与（AND，符号 `&`）：同 1 为 1。
     (b) 或（OR，符号 `|`）：有 1 为 1。
     (c) 异或（XOR，符号 `^`）：不同为 1。
  2. 另外需要知道异或的常用性质：
     (a) $x \oplus x = 0$。
     (b) $x \oplus 0 = 0$。
     (c) $x \oplus y = y \oplus x$。
     结合性质可以对异或公式进行简化，如：

$$x \oplus y \oplus x \oplus y \oplus z = z$$

字符串的子串/子序列

- **题目**
  2022 CSP-J 第14题
  2017 NOIP普及组 第14题
  2012 NOIP普及组 第19题
- **经验**
  1. 注意区分子串与子序列的区别：**子串连续，子序列不连续。**
  2. 一个 n 长串一共有 $C_n^2 + C_n^1 + 1 = \frac{n \times (n+1)}{2} + 1$ 个子串（包含空串），2^n 个子序列（包含空序列）。
  3. 如果需要去重（比如问子串/子序列种类），长度不长可考虑暴力穷举。

容器进出顺序

- **题目**
  2022 CSP-J 第5题
  2022 CSP-J 第2题

2021 CSP-J 第5题
2017 NOIP普及组 第16题
2015 NOIP普及组 第15题
2013 NOIP普及组 第7题
2012 NOIP普及组 第2题
2012 NOIP普及组 第12题

- **经验**
  1. 虽然题型可能会有很多种形式，但一般按照容器概念模拟即可：
     (a) 队列：先进先出（排队）。
     (b) 栈：先进后出（弹匣）。

IP地址类型

- **题目**
  2014 NOIP普及组 第12题
  2013 NOIP普及组 第13题
- **经验**
  1. 第一组数二进制与地址类的对应关系：
     (a) `0…`：A 类。
     (b) `10…`：B 类。
     (c) `110…`：C 类。
     (d) `1110…`：D 类。
     (e) 其他：E 类。