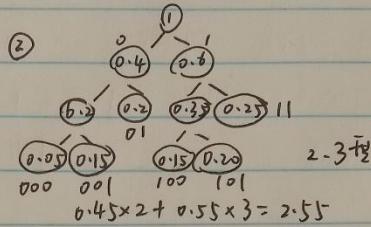
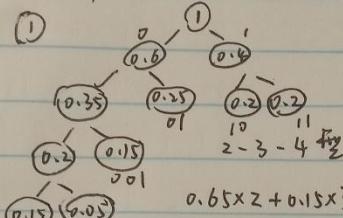


$$18. \underline{0.05+0.15=0.20} \quad H = -0.25 \times \log_2 0.25 + 2 \times (-0.2) \times \log_2 (0.2) + 2 \times (-0.15) \log_2 (0.15) + (0.05) \times \log_2 (0.05) \\ = 2.466$$

有两种霍夫曼树：



两种霍夫曼编码的平均长度相同 空余度： $1 - \frac{2.466}{2.55} = 0.033$

19. (1) 因为每次进行函数调用时，都会存在一些压栈操作，比如 `push`、`pop` 都会被存在堆栈里，还有一些函数参数也会。如果一直在发生调用而没有存在返回，那么就没有了单栈，堆栈得不到释放，造成栈溢出。

(2) 增加栈空间大小；使用动态内存分配；减少局部变量使用

$$20. a_0(F_1) \quad 9. (1) \text{ 有符号: } -2^{19} \sim 2^{19}-1$$

$$to(F_1) \quad (2) \text{ 有符号: } -2^{16} \sim 2^{16}-1 = -2048 \sim 2047 \text{ 范围}$$

$$so(F_2) \quad (3) \text{ 可以, } jalr 操作为 rd = pc + 4; pc = rs1 + imm,$$

$$a_0(F_2) \quad lui 操作为 rd = imm << 12 \text{ 其中 imm 为高 20 位}$$

$$a_1(F_2) \quad \text{可以先用 lui 将绝对地址高 20 位存到 rs1 中, 然后 jalr 时将存有高 20 位的寄存器放在 rs1, 低 12 位放在 imm, 从而实现任意 32 位绝对地址跳转。}$$

$$ra(F_2) \quad \text{位移量放在 imm, 从而实现任意 32 位绝对地址跳转。}$$

$$to(F_2) \quad \text{地址跳转。}$$

$$t_1(F_2)$$

10. 32位后序为16位的条件：① 地址或偏移量较小 ② 有一个寄存器为 x_0 、 x_1 或 x_2 位

③ 目的寄存器与一个源寄存器相同 ④ 使用的寄存器是 8 个最常用的

不是。CR、CI 和 CS 可使用所有的，但 CIW、CL、CS、CB 只能用其中的 8 个