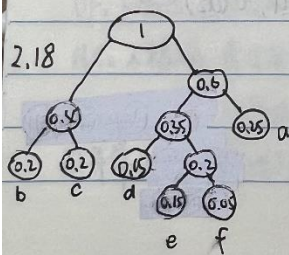


ret

2.9 (1) $PC - 2^{19} \sim PC + 2^{19} - 1$ (2) $PC - 2^{11} \sim PC + 2^{11} - 1$
 (3) 能, 可先用 lui 将地址高 20 位存入 rs 寄存器中, 再将低 12 位地址直接作为 jatr 指令, 从而完成 $PC = \text{rs} + \text{imm}$ 跳转

2.10 一般可被压缩的 32 位指令功能为普通的算术、逻辑、移位、比较、分支、访存等
 RVC 中许多指令 R 能访问 8 个整数寄存器 ($\text{x8} - \text{x15}$), 如数据存取与算术运算指令, 不过也有指令可使用所有通用寄存器作为操作数



$$H = - \sum_{i=1}^6 p_i \cdot \log_2 p_i = 2.466$$

$$\text{平均长度 } L = (0.25 + 0.2 + 0.2) \times 2 + 0.15 \times 3 + (0.15 + 0.05) \times 4 = 2.55$$

$$R = 1 - \frac{2.466}{2.55} = 0.033$$

2.19 (1) 每在已有函数未执行完时递归调用新函数时都将在现有栈上维护一个新的栈层, 嵌套过多时可能造成栈溢出

(2) 如对于递归调用死循环的栈溢出, 可以改写代码使函数检测异常情况;

对于栈过长的情况也可动态分配内存存放变量

2.20 $\text{ra}(\text{F1})$

$\text{t0}(\text{F1})$

$\text{ra}(\text{F2})$

$\text{s0}(\text{F2})$

$\text{t0}(\text{F2})$

$\text{t1}(\text{F2})$