

3.28 第六周

9. (1) 可以跳转到  $(PC_{\text{当前值}} - 2^{19})$  到  $(PC_{\text{当前值}} + 2^{19} - 1)$  的范围。

(2) 可以跳转的范围是  $(PC_{\text{当前值}} - 2^{11})$  到  $(PC_{\text{当前值}} + 2^{11} - 1)$  即  $(PC - 2048 \sim PC + 2047)$ 。

(3) 不可以。因为 `lui` 指令将一个 16 位的立即数左移 16 位，然后存储到目标寄存器的高 16 位，只能跳转到一个与 `0x FFFF0000` 按位与的目标地址，因此目标地址低 16 位都是 0。因此使用 `jalu` 指令时只能读取到 `lui` 指令存储的高 16 位地址。

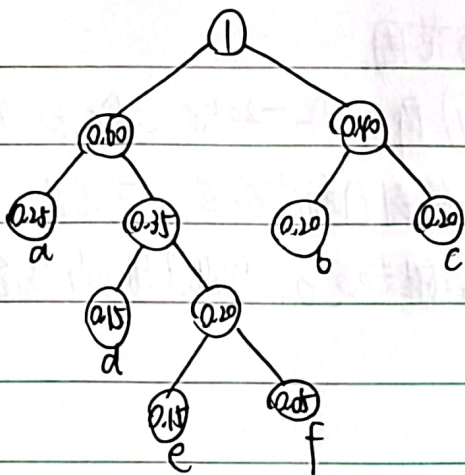
10. 每一条常用的 32 位指令被压缩为 16 位 RVC 指令的条件：

该指令只使用 16 位内的操作数；该指令的立即数不超过 6 位（或 12 位，具体取决于指令类型）；

该指令的目标寄存器是 8 个低位寄存器（X0 至 X7）之一。



18.



$$\text{平均长度} = \sum_{i=1}^6 p_i l_i = 2.15$$

$$\text{信息冗余度} = 1 - \frac{\sum_{i=1}^6 p_i \log_2 p_i}{\log_2 6} = 0.032958$$

19. 在程序中, 函数调用会在栈中创建一个新的栈帧, 用来存储函数的参数、局部变量、返回地址等信息。函数执行完成后, 该函数的栈帧会被弹出。因此当函数嵌套调用层数过多时, 栈中创建的栈帧数量也随之增加。如果栈空间不足, 就会发生栈溢出错误。

缓解或避免方法: 1. 优化算法以减少递归深度;

2. 使用循环代替递归;

3. 增大栈空间大小。

4. 使用尾递归优化, 在函数的最后一步调用自身。

5. 使用动态规划算法。



20.	栈地址	内容
		ra(F1)
	sp+4	so(F1) 参数1
	sp+8	to(F1)
		ra(F2)
	sp+4	so(F2) 参数1
	sp+8	si(F2) 参数2
	sp+12	to(F2)
	sp+16	t1(F2)

