

## 第六周

9. A: (1) jal 指令包含 20 位有符号立即数编码,

相较当前 PC 可跳转的地址空间为  $-2^{19} \sim 2^{19}-1$

(2) B 型指令包含 12 位有符号立即数编码,

可跳转至  $-2048 \sim 2047$

(3) 先用 lui 将绝对地址的高 20 位装载到寄存器 t0 中,

再用 jalr 指令, 通过偏移量补全低 12 位

即完成 32 位绝对地址的跳转

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

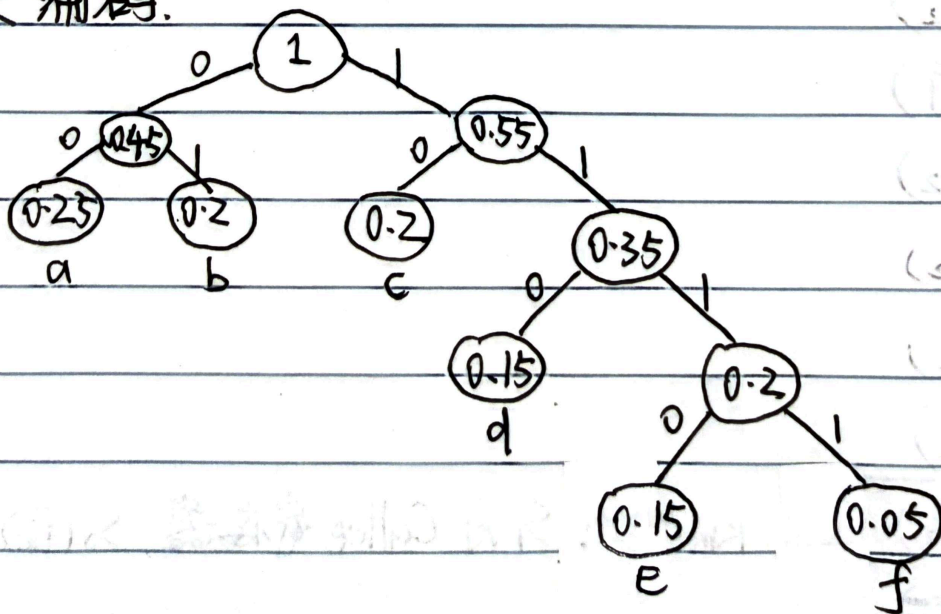
① 立即数足够小, 或

② 其中一个寄存器是 X0, X1 或 X2, 或

③ 目的寄存器与第一个源寄存器相同, 或

④ 使用的寄存器为 8 个最常用的寄存器.

18. A: 霍夫曼编码.



编码:

$a_i$	$P_i$	code	length
a	0.25	00	2
b	0.2	01	2
c	0.2	10	2
d	0.15	110	3
e	0.15	1110	4
f	0.05	1111	4

平均长度:  $\bar{L} = \sum_{i=1}^6 L_i P_i = 2.55$

信息熵:  $H = -\sum_{i=1}^6 P_i \log_2 P_i = 1.709$

冗余度  $R = \frac{\bar{L} - H}{\bar{L}} = 33.0\%$

19. A: (1) 函数嵌套调用层数过多时, 每一层函数的局部变量、返回地址、callee 变量等都会保存在栈中, 可能超出栈分配的空间, 导致栈溢出.

(2) 解决方法:

尽量避免使用递归; 对大型局部变量分配动态空间.

20. A:  $ra(F_1)$

$a_0(F_1)$

$t_0(F_1)$

$ra(F_2)$

$s_0(F_1)$

$a_0(F_2)$

$a_1(F_2)$

$t_0(F_2)$

$t_1(F_2)$

$ra(F_3)$

Rm:  $s_0, s_1$  为 Callee 寄存器,  $s_0(F_2)$  在  $F_3$  调用后保存.