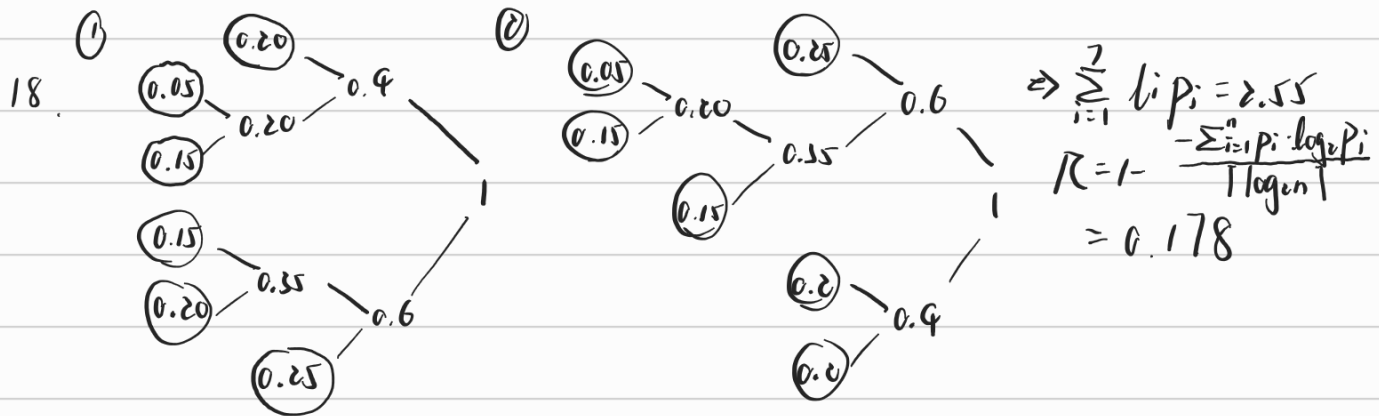


第九周 . 9.10.18.19.20

9. (1) JAL为2字节对齐 $\Rightarrow 2 \times 2^{20} = 2MB \Rightarrow$ 范围 $\pm 1MB$.
 (2) 每行分支同样2字节对齐 $\Rightarrow 2 \times 2^{12} = 8KB \Rightarrow$ 范围 $\pm 4KB$
 (3) 可以在32位系统中, 一条LUI指令将目标地址的高20位加载到rs1中, JALR指令可以加上低12位.

10. 符合以下事件: ① 立即数或地址偏移量较小时 ② 其中一个 reg 为 zero, 0B1 链接寄存器 (x1) 或 ABI 栈寄存器 (x2) ③ 目标寄存器和第一个源寄存器相同 ④ 只使用了 0~7. 8个 reg (只需要三位表示)



19. (1) 栈为先进后出的数据结构, 与函数递归调用过程类似, 递归的每一层中的局部变量地址等都会写入栈中, 当嵌套调用层数过多 (如死循环中) 占用的栈空间过多直至耗尽手动或自动分配的空间就会 overflow
 (2) 避免 overflow 的方法有: ① 限制最大递归深度 ② 增加栈空间
 ③ 手动管理栈, 进行空间调配和释放 ④ 尾递归优化 ⑤ 用循环代替.

20.

ra (F1)	:	t0 (F2)
arg1 (F1)	:	t1 (F2)
t0 (F1)	:	s0 (F2)
s0 (F1)	:	s1 (F2)
ra (F2)		
arg1 (F2)		
arg2 (F2)		

(Note: In the original image, there is a blue arrow pointing from 'ra (F2)' to 'ra (F1)' with the text '保持F1' (Keep F1).)