

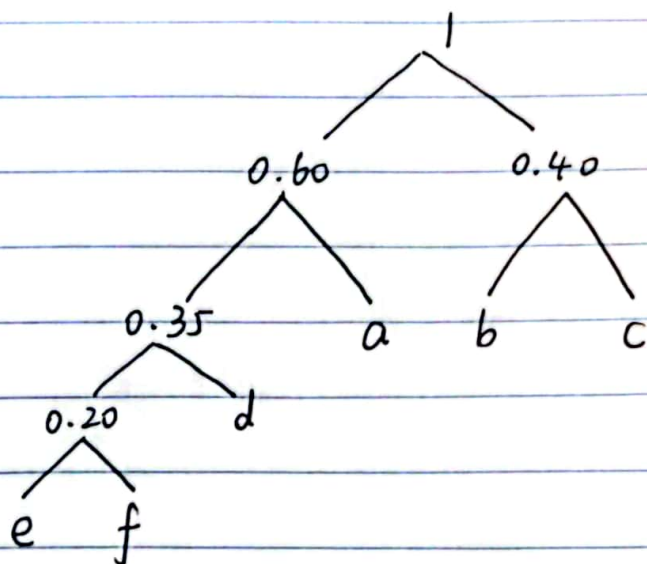
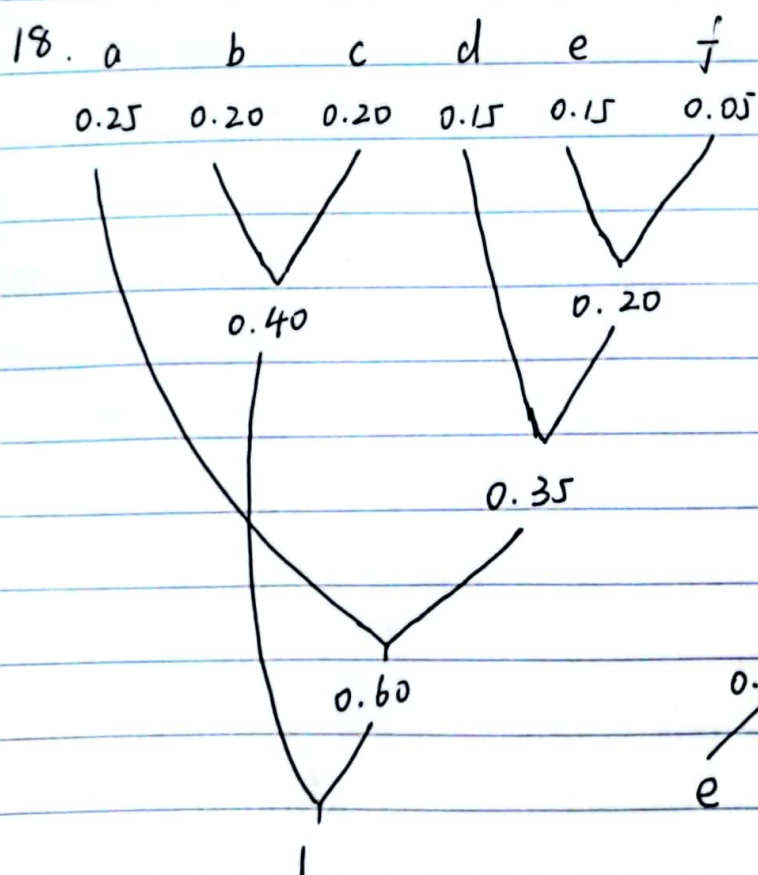
9.10. ~~11.12~~.18.19.20.

9. jal 指令可以跳转范围为  $PC + 1 - 2^{20}$  的地址范围

12) 分支指令可以跳转到地址范围为  $PC + 1 - 2^{12}$

13) lui 指令可以与一条 jalr 指令组合使用, 将低 16 位指定为立即数, 并使用寄存器来指定高 16 位. 这样的跳转操作并不是“任意”的, 因为目标地址仍受到寄存器的限制.

10. 条件: 指令必须是支持 RVC 压缩的指令类型之一 (即 L, S, B, I, J, R, F, D, RV64C); 只能包含指定的寄存器 (x8-x15) 和立即数  $[-2048, 2047]$ , 不能使用其他寄存器或更大的立即数; 不能使用 load/store 指令的缩写格式, 不能包含特权指令, Fence 指令或系统调用指令. 都可以.



- a ~~00~~ 01 操作码平均长度
- b 10  $\sum_{i=1}^6 P_i l_i = 2.55$
- c ~~000~~ 11  $H = - \sum_{i=1}^n P_i \cdot \log_2 P_i = 1.55 + 2.466$
- d ~~0000~~ 001 冗余度
- e ~~0000~~ 0000  $\frac{2.55 - 2.466}{2.55} = 3.3\%$
- f ~~0000~~ 0001

19.

解: 递归函数会不断地调用自身, 每次调用都会栈中保存一份函数的上下文信息, 如果递归调用层数过多, 栈空间会不断地增长, 最终超出栈的容量限制, 导致栈溢出错误。解决: 增加栈空间, 减少函数递归。

20.  $ra(F1)$   $to(F1)$   
 $ra(F2)$   $so(F1)$   
 $to(F2)$   
 $t1(F2)$   
 $so(F2)$   
 $s1(F2)$

