

9. (1) 第1位是符号位, 故可跳转范围是 $-2^{19}+1 \sim 2^{19}-1$

(2) 同理, 范围是 $-2^{18}+1 \sim 2^{18}-1$

(3) 可以, 因为 lui指令的目标是将高20位加载到某个寄存器中, 低12位补充零, 再用 jalr指令将绝对地址的低12位填上即可

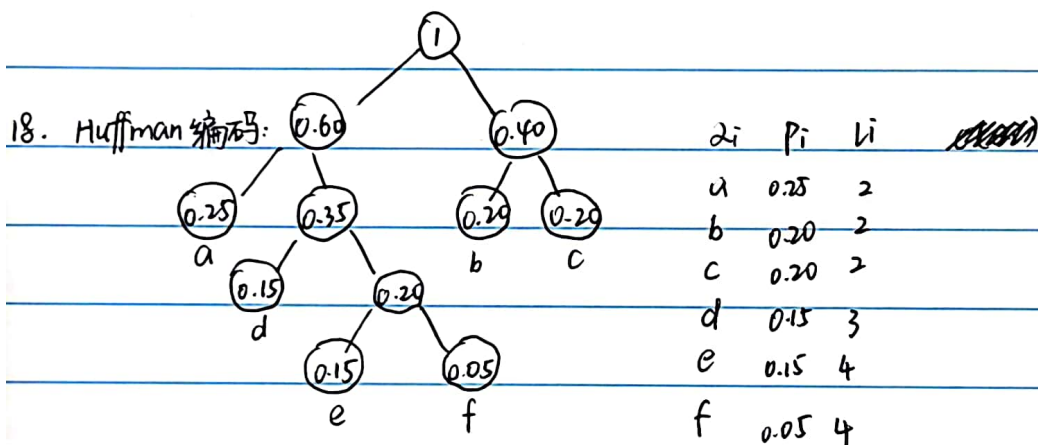
例: lui x1, 0x12345

jalr x0, x1, 0x678 即可把地址跳转到 0x12345678' 绝对地址处.

10. 满足条件: ① 指令只涉及到通用寄存器的使用, 不涉及特殊的寄存器或内存操作

② 指令的立即数或偏移量在特定的范围内

不是所有指令都可以使用完整的32个通用整型寄存器, 例如 Load和 Store指令只能用8个通用寄存器.



平均长度 $\bar{l} = \sum_{i=1}^7 l_i p_i = 2.55$ $H = -\sum p_i \log_2 p_i = 2.47$

$R = 1 - \frac{2.47}{2.55} = 3.3\%$

19. (1) 例: 死循环导致栈溢出 long func(int n)

```

{ return n * func(n-1);
}

```

由于没有定义一个实在的值, 会陷入死循环中, 递归函数没有出口, 每次都把返回地址压入栈中, 导致溢出错误

(2) 修改程序, 使它没有死循环; 修改栈的大小, 避免溢出

20. $ra(F_1)$

$to(F_1)$

$so(F_1)$

$ra(F_2)$

$ao(F_2)$

$ai(F_2)$

$to(F_2)$

$ti(F_2)$

$so(F_2)$

$si(F_2)$