

-0.5

-11 46 43 856

9. (1) 除去符号位 20位的立即数可表示  $-2^{19}$  到  $2^{19}-1$  之间的数值, 在 risc-v 中须将偏移量左移一位再与 PC 高 12 位, PC+4 和符号位相拼接, 因此 "jal" 指令可以跳转到相对于当前 PC  $\pm 1$  MB 范围的任何地址。

(2) 指令跳转范围为  $\pm 4$  KB 地址空间

(3) risc-v 中可以使用一条 lui 指令和 jalr 指令的组合完成任意 32 位绝对地址的跳转操作, lui 指令可以将一个 20 位的立即数左移 12 位, 得到一个 32 位的数, 存放在目标寄存器的高 20 位中, 故 lui 指令可以用于加载一个 32 位地址的高 20 位, jalr 指令可以将指令地址加上一个 12 位的立即数, 并将结果存放在目标寄存器中, 同时将 PC 设置为当前指令地址加上一个偏移量, 结合这两条指令, 具体代码实现如下:

```
lui x1, %hi()
```

```
addi x1, x1, %lo()
```

```
jalr x0, x1, 0
```

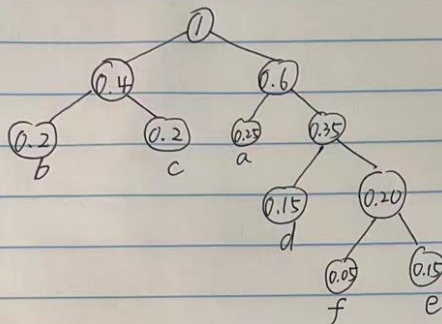
10. 在 RVC 指令集中, 可以将常用的 32 位指令压缩为一条 16 位的 RVC 指令条件为:

1. 指令没有使用特权指令
2. 没有使用 64 位寄存器
3. 指令操作的立即数是 16 位或更小的
4. 指令没有浮点操作码

在 RVC 指令集中, 所有指令都可以使用完整的 32 个通用寄存器, 但不支持浮点计数器的操作。

-0.5  
-1.4643856

18. a: a b c d e f  
 $P_i$  0.25 0.20 0.20 0.15 0.15 0.05



$d_i$	$c(a_i)$	$l_i$	$P_i$	平均长度: $L = \sum P_i l_i = 2.55$
a	10	2	0.25	信息冗余度:
b	00	2	0.2	$R = \frac{\sum_{i=1}^n P_i \cdot \log_2 P_i}{\log_2 n}$
c	01	2	0.2	$H = - \sum_{i=1}^n P_i \log_2 P_i = 2.46596$
d	110	3	0.15	信息冗余度 = $1 - \frac{H}{L} = 3.296\%$
e	1111	4	0.15	
f	1110	4	0.05	

19. 如果函数嵌套调用的层数过多, 栈空间可能会被耗尽, 这种情况下, 新的函数调用将无法分配所需的栈空间, 从而导致栈溢出。当栈溢出时, 程序将崩溃, 并显示一个错误信息。

2) ① 增加栈空间

② 减少函数调用嵌套层数

③ 检查代码中是否存在无限空循环、递归调用等问题

④ 使用动态内存分配, 但要注意及时释放已分配的内存。



-0.410

20.

ra (F1)

to

so

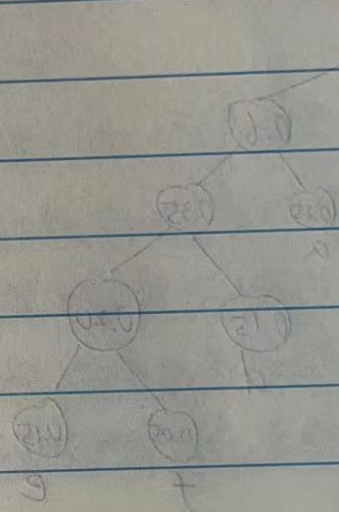
~~ti~~ ra (F2)

~~so~~ to

ti

so

si



22.5 = 11.5, 22.5 = 11.5

22.5 = 11.5, 22.5 = 11.5

22.5 = 11.5, 22.5 = 11.5

22.5 = 11.5, 22.5 = 11.5

22.5 = 11.5, 22.5 = 11.5

22.5 = 11.5, 22.5 = 11.5

22.5 = 11.5, 22.5 = 11.5