9.

(1) 20位有符号数可以覆盖 2^{20} 个字节，由于 RISCV 64 中一条指令占4个字节，所以可以覆盖从当前指令开始的向上 2^{17} 条和向下 $2^{17}-1$ 条指令的地址空间

会被扩展为21位有符号数吗？

(2) 与上题同理，除去一位符号位，可覆盖从当前指令开始的向上 2^9 条和向下 2^9-1 条地址空间

(3) 可以。一条 lui 指令可以在寄存器中存入32位数的高20位，而 jalr 恰好可以取出该数后再加上12位组成32位地址并跳转。

是所有数位置0？四字节对齐？

10.

(1) 能够被压缩为16位 RVC 指令的条件是

① 立即数或地址偏移量很小

② 使用的寄存器中有更寄存器 (x0) 或 ra 寄存器 (x1) 或 sp 寄存器 (x2)

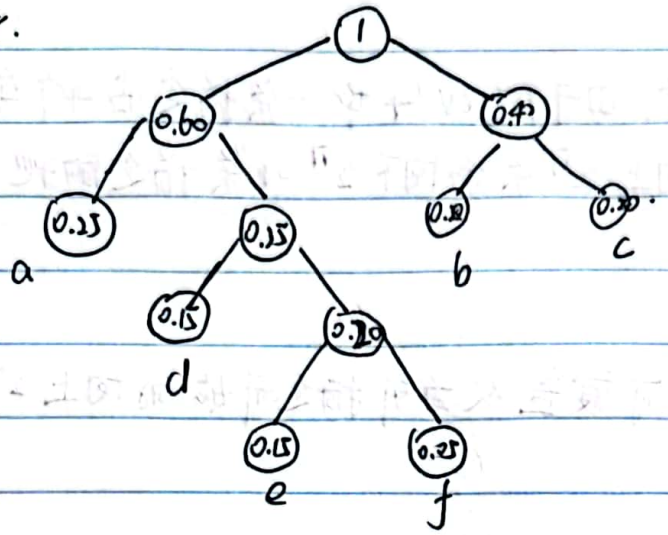
③ 目标寄存器和第一个源寄存器相同

④ 使用的寄存器是最常用的8个 (x8 - x15).

以上条件满足之即可

(2) RVC 指令不同类型的指令都会使用32个通用寄存器中的一部分，但是没有某个类型的指令可以使用全部寄存器

18.



∴ 编码为 a 00

b 10

c 11

d 010

e 0110

f 0111

平均码长 $= \sum_{i}^{6} p_i l_i = 2.55$

最短平均码长 $H = -\sum_{i}^{6} p_i \cdot \log_2 p_i$

≈ 2.466

∴ 信息冗余度 $= 1 - \frac{2.466}{2.55}$

$= 3.3\%$

19.

(1) 递归函数每一次都会调用自己作为子函数，而调用函数时，就会执行函数开始时的初始化工作，会将栈指针sp向下移动一段距离(如32.48)，开辟一块栈空间供本次函数运行使用。

一般情况下，在函数结束时才会释放栈的这部分空间。但是递归函数会在结束前调用子函数，必须在递归结束后，才可以将每次调用开辟的空间依次释放。这样做也是为了父函数的数据不被覆盖，防止出现找不到返回地址无法返回等问题

所以递归函数死循环或调用次数太多，只开辟栈空间不释放，就有可能将栈的空间全部用完，出现栈溢出的问题

(2) 方法

① 改变函数实现方法，如不使用递归改用循环结构.

② 减少每次开辟的栈空间，只开辟必要的空间，可以增加固定栈总空间情况下可以调用的次数

③ 减少代码中使用栈的变量，更多使用寄存器存储变量，可以进一步压缩需要开辟的栈空间

④ 将一些变量改为常量，也可以压缩需要的栈空间.

⑤ 在不确定是否会溢出的时候，HSE RZSC-V 可以用栈指针边界检查，(用一条指令检查栈指针位置) 或栈空间限制 (设置一个比内存小的最大栈大小) 来防止栈指针越界，在栈溢出前结束程序的运行.

12.

首次调用F3前：ha (F1)

s0 (F1)

t0 (F1) (参数)

ha (F2)

s0 (F2)

t0 (F2) (参数)

t1 (F2) (参数)

s1 (F2) (可能有的计算结果，因为F2会返回一个int值)