2020年重庆大学917计算机考研真题参考答案

一、选择题

1、A
2、A
3、D
4、A
5、B
6、D
7、D
8、B
9、B
10、A
11、C
12、C
13、D
14、C
15、C
16、C
17、D
18、C
19、C
20、D
21、C
22、D
23、A
24、D
25、B
26、B

二、解答题

1、解

右移：$1.000_2*2^{-1}+ -0.111_2*2^{-1}$

相加：$1.000*2^{-1} + -0.111*2^{-1} = 0.001*2^{-1}$

规格化，检查有无溢出 $1.000*2^{-4}$，没有溢出

舍入：无，$1.000*2^{-4} = 0.0625$

2、解

I-cache miss rate 3%
D-cache miss rate 7%
} Miss penalty = 120 cycles } 比例 75%

Base CPI = 2.

(1) 指令缺失时钟周期 = I × 3% × 120 = 3.6I
数据缺失时钟周期 = I × 75% × 7% × 120 = 2.1I
总的 CPI = (3.6I + 2.1I)/I = 5.7I/I = 5.7
实际 CPI = (5.7I + 2I)/I = 7.7I/I = 7.7

(2) 降低 1% ⇒ I-miss 1.5%
D-miss 3.5%
} Miss penalty = 150 cycles

指令缺失: I × 1.5% × 150 = 2.25I
数据: I × 3.5% × 75% × 150 = 1.3125I
实际 CPI = (2.25I + 1.3125I + 2I)/I = 5.5625 < 7.7
果然更加优秀.

3、解
加法执行过程如下:
(1) PC 寄存器中地址送指令存储器，并读指令;PC 地址同时送地址加法部件实现 PC+4;
(2)根据指令中的寄存器的 rs 和 rt 的编号从寄存器堆中读出两个寄存器中的值;
(3)rs 和 rt 中的值送入运算器中进行加法运算;
(4)运算的结果送入寄存器堆中的 rd 寄存器;
(5) 将 PC+4 的值送入 PC
该指令执行的关键路径为:
取指令->取操作数(同时控制器译码)->多路选择器->运算器->多路选择器->多路选择器->写寄存器堆
因此总的时间为:400ps+200ps+30ps+120ps+ 30ps+30ps+200ps=1010ps

4、解：答案存疑

5、解：在数据第一次写入到某个存储位置时，首先将原有内容拷贝出来，写到另一位置处，然后再将数据写入到存储设备中，该技术只拷贝在拷贝初始化开始之后修改过的数据。

6、解：

（1）FCFS:P1->P2->P3->P4

SJF:P1->P2->P4->P3

(2)9.25,9

7、解：这个题，注意题目说：刚完成 69，现相应 34 号

（1）SSTF：34，23，11，0，47，50，56，89，90，160

平均寻找长度：（34+160）/9

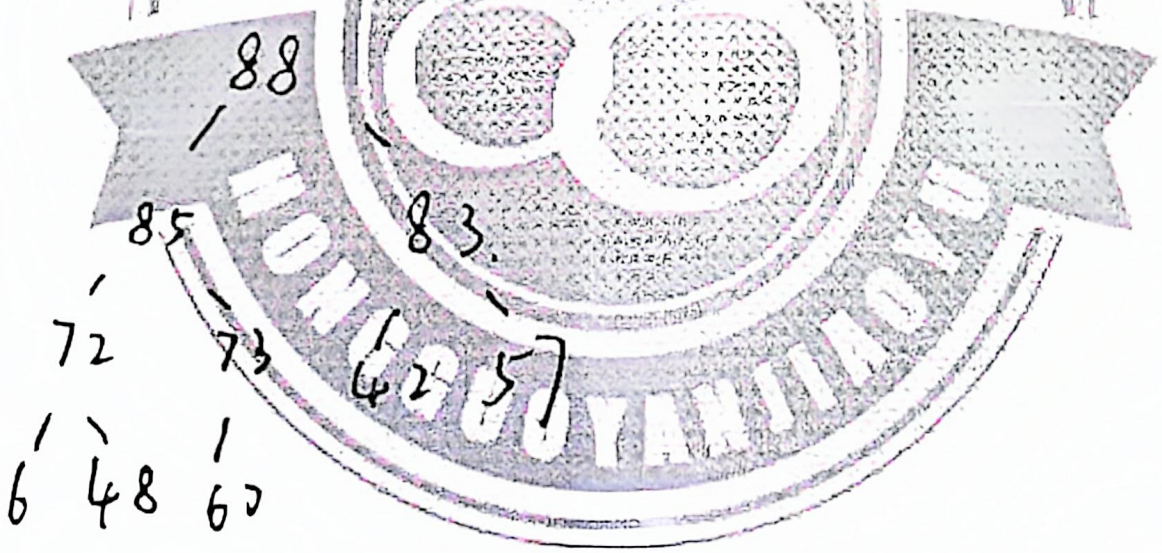
（2）69->34 是从大往小的方向移动

8、解：该题和以往不一样，是用 C++语言作答。

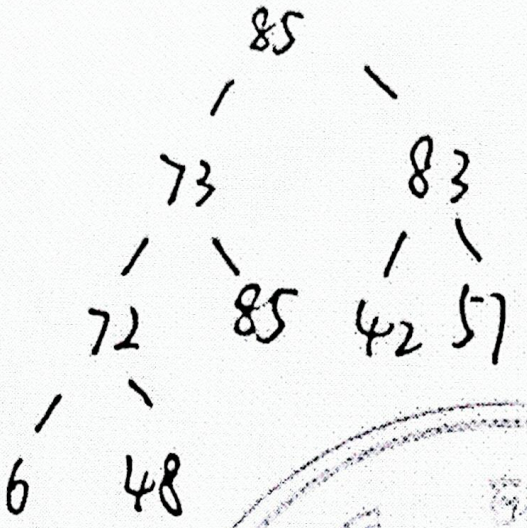
不过也不是很难，有一点 C++的基础应该就可以做这个题目。

9、解：显然可得，该功能为中序遍历。

10、 解：（1）



（2）去掉最顶那个值，然后把最下一排，最右边哪一个移到最顶上的。



11、 解：同 2011 年 41 题。若有疑问可以问。

41. 解析：先计算每个 key 的 H(key)

key	19	1	23	14	55	20	84	27	68	11	10	77
H(key)	6	1	10	1	3	7	6	1	3	11	10	12

地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
key	27	1	14	55	68	84	19	20		10	23	11	77						
冲突次数	2	0	1	0	1	2	0	0		2	0	0	0						
比较次数	3	1	2	1	2	3	1	1		3	1	1	1						

$$ASL_{成功}=\frac{12+(2+1+1+2)}{12}=\frac{5}{3}$$

12、 解：

(1) 通信使用明文可能会被窃听

按 TCP/IP 协议族的工作机制，互联网上的任何角落都存在通信内容被窃听的风险. 而 HTTP 协议本身不具备加密的功能，所传输的都是明文. 即使已经经过过加密处理的通信，也会被窥视到通信内容，这点和未加密的通信是相同的. 只是说如果通信经过加密，就有可能让人无法破解报文信息的含义，但加密处理后的报文信息本身还是会被看到的.

(2) 不验证通信方的身份可能遭遇伪装

在 HTTP 协议通信时，由于不存在确认通信方的处理步骤，因此任何人都可以发起请求. 另外，服务器只要接收到请求，不管对方是谁都会返回一个响应. 因此不确认通信方，存在以下隐患：

无法确定请求发送至目标的 Web 服务器是否是按真实意图返回响应的那台服务器. 有可能是已伪装的 Web 服务器；无法确定响应返回到的客户端是否是按真实意图接收响应的那个客户端. 有可能是已伪装的客户端；无法确定正在通信的对方是否具备访问权限. 因为某些 Web 服务器上保存着重要的信息，只想发给特定用户通信的权限；无法判定请求是来自何方、出自谁手；即使是无意义的请求

也会照单全收，无法阻止海量请求下的 DoS 攻击；

(3) 无法证明报文完整性，可能已遭篡改

所谓完整性是指信息的准确度．若无法证明其完整性，通常也就意味着无法判断信息是否准确．HTTP 协议无法证明通信的报文完整性，在请求或响应送出之后直到对方接收之前的这段时间内，即使请求或响应的内容遭到篡改，也没有办法获悉．

比如，从某个 Web 网站下载内容，是无法确定客户端下载的文件和服务器上存放的文件是否前后一致的．文件内容在传输途中可能已经被篡改为其他的内容．即使内容真的已改变，作为接收方的客户端也是觉察不到的．像这样，请求或响应在传输途中，遭攻击者拦截并篡改内容的攻击称为中间人攻击(Man-in-the-Middle attack，MITM)．

13 解：

(1) L1:2^(32-21)-2=2^11-2=2046

L2:2^(32-20)-2=2^12-2=4094

L3:2^(32-22)-2=2^10-2=1022

L4:2^(32-22)-2=2^10-2=1022.

(2) 见下表

R1：记得聚合

目的地址	下一跳	发送端口
202．20.12.0/22	R2	P1
2020.20.0.0/19	R2	P1

R2：记得聚合

目的地址	下一跳	发送端口
202．20.12.0/22	-	P3
2020.20.0.0/19	-	P2

R3：记得聚合

目的地址	下一跳	发送端口
202．20.12.0/22	R2	P1
202.20.0.0/21	-	P2
202.20.16.0/20	-	P3
202.20.8.0/22	-	P4

(3) 最长匹配前缀，P2