上 海 交 通 大 学 试 卷（ A 卷）

（ 2022 至 2023 学年 第 1 学期 ）

班级号______________________ 学号______________ 姓名 ____________

课程名称_______计算机系统工程_______成绩 _____________

Problem 1. Transaction (26’)

1. Ross uses MVCC (Snapshot Isolation) to execute transactions. Assuming the Initial value at t = 0 is A: (A0, A5, A10) and B: (B0, B7, B15). Given the schedule below, please answer the following questions:

Time	20	21	22	23	24	25	26	27	28
T1	S	R(A)	R(B)		W(A)	W(B)	C		
T2				S		R(A)	R(B)		C
T3		S	R(A)		W(B)	C			
T4			S	R(B)	W(A)			C	

S: Start; R: Read; W: Write; C: Commit

   a. What value will T2 get at time 25 and time 26? (2')

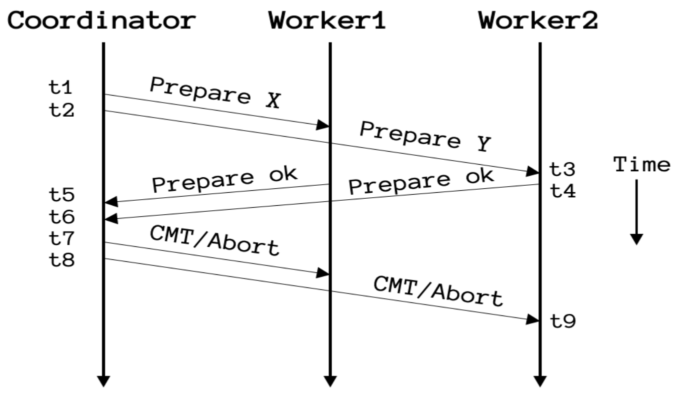
   b. Which transactions will abort and which will commit? (2')

   c. What are the final values of A and B after T1-T4 commit ? (i.e., after time 28) (2')

   d. Is this schedule serializable? Please briefly explain your reasons. (4')

   e. How to generate timestamps (i.e., the start timestamp and commit timestamp) under a single machine setting and a distributed setting, respectively? (4')

2. Ross tries to utilize 2PC protocol to ensure multi-site atomicity.



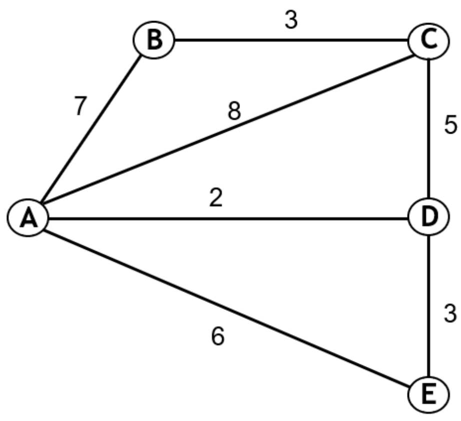
a. If the Coordinator crashes at t (t7 < t < t8), can worker2 commit or abort its sub-transaction? Please briefly explain your reasons. (4')

b. If worker1 replies **not ok**, worker2 replies **ok** at the PREPARE phase, and worker2 crashes right after t4, what should worker2 do after it restarts? (Hint: worker2 may recover before or after t9) (4')

c. Please explain why 2PC has low availability under machine crashes, and give a possible solution. (4')

Problem 2. Network (24')

1. Consider the following network topology using **link-state routing**. Please write down advertisement content of node C and node E. Please write down node A's routing table with minimum routing costs. (8')



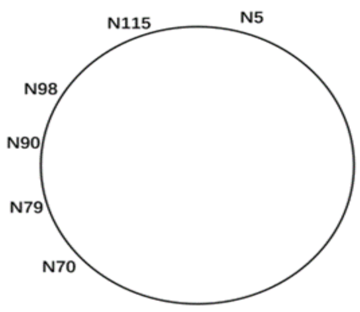
C's advertisement: ______________________________

E's advertisement: ______________________________

A's routing table:

dst	route	cost
B		
C		
D		
E		

2. Suppose there is a DHT with hash space [1,120]. There are 6 nodes: Node 5, Node 70, Node 79, Node 90, Node 98 and Node 115. Each key is stored at its successor (node with next higher ID) using consistent hashing. If a node crashes, its direct successor will be responsible for its keys (e.g., if Node 70 crashes, Node 79 will be responsible for its keys). The DHT also uses finger tables, which is shown below. Please answer the following questions.



Node	Finger table
5	[70]
70	[79, 90, 115]
79	[90, 98, 115, 70]
90	[98, 115, 5, 70]
98	[115, 70]
115	[5, 70]

a. Please write the lookup process for **Key 89** starting from **Node 5**. (Please write in the following form : N5->Nx->Ny->…) (4')

b. If every node only stores the finger table, and both Node 79 and Node 90 crash, will the previous lookup succeed? Please briefly describe how DHT handles such a failure. (4')

3. When transmitting bits, Parity bits are added in order to detect and correct errors. Suppose a Hamming (7,4) code that codes 4 transmitted bits into 7 bits with the following equations (note the "%2" part):

$$P_1 = (P_3 + P_5 + P_7)\%2$$
$$P_2 = (P_3 + P_6 + P_7)\%2$$
$$P_4 = (P_5 + P_6 + P_7)\%2$$

a. How many bits of error can be corrected? Why? (4')

b. If the 7 bits are 0110001, which bits are incorrect? Why? (4')

Problem 3. Replication (22’)

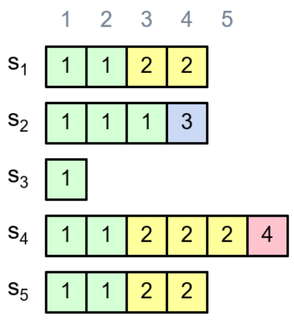
Here is the pseudocode for **Paxos**. We assume that all the servers are both acceptors and proposers.

```
                              States
V: the chosen value of the proposer
M_n: my proposal number
N_h: highest proposal number seen
N_a: highest proposal number accepted
V_a: accepted value of Na

  -------------------------------------------------------------
                              Proposer
  propose(v):
  1  choose M_n > N_h
  2  send <proposal, M_n > to all nodes
  3  If gets promise-ok from a majority:
  4     If V != null, V = the value of the highest N_a received
  5     If V = null, then can pick any V
  6        send <accept, M_n, V> to all nodes
  7        If Leader gets accept-ok from a majority:
  8           (a)__________________________________
  9        If Leader fails to get majority accept-ok
 10           (b)__________________________________

  -------------------------------------------------------------
                              Acceptor
  acceptor's <proposal,N > handler:
  1  If N < N_h
  2     (c)______________________
  3  Else
  4      N_h=N; reply <promise-ok, N_a, V_a>

  acceptor's <accept,N,V> handler:
  1  If  N < N_h
  2      reply <accept-reject>
  3  Else
  4      (d)_______________________
  5      reply <accept-ok>
```

1. Please fill in the blanks in the above pseudocode. (4')

2. The pseudocode requires persisting **Nh**, **Na** and **Va** on disk. Ben decides it may be sufficient just to store Nh and Va on disk, and if a node reboots, the node sets Na to the saved Nh value. Is this modified Paxos protocol correct? If so, why? If not, please give an example to explain your answer. (6')

3. Consider the following scenario in a **Raft** Group. Each box represents one log entry. The number in the box is its term. The content of the log entries are omitted for simplicity. Please answer the following questions.



   a. If S4 crashes and never reboots, which servers can be elected as the leader of term5? Explain why. (3')

   b. Which log entries may safely be applied to state machines? Please explain why other logs can not be applied. (3')

4. What is CAP theorem? Which property of CAP is sacrificed in Raft? Please briefly how it fails to achieve so. (6')

Problem 4. Security (28')

```
void getLine(char *buf)
{
    while(true){
      char c = getchar();
      if (c == '\n')
          break;
      *buf++ = c;
    }
}
```

```
void login()
{
    char pwd[32];
    getLine(pwd);
    if   (pwdCheck(pwd))
          return   true
    return false
}
```

1. Could you briefly explain one vulnerability of the above program? (4')

2. Please give at least two techniques learnt from the class to mitigate the above vulnerability. (4')

3. Taint tracking can track the lifecycles of sensitive data. In the following program, "i" is tainted. Please fill the taint table for **all the variables** *after* the program ends. (Hint: you need to consider different cases.) (6')

```
i = input()
a = 2
if i % 3 == 0:
    j = i * a
else:
    j = 2**a
k = j + a
```

Variable	Taint status (True / False)
i	
(you can add more lines)	

4. Taint tracking is not 100% safe, it can be bypassed. Please write a function to remove taint from any variable. (4')

```
uint16 remove_taint(uint16 v)
{
    Your code here.
    You don't have to write all the code if it's too long.
}
```

5. Shadow Stack and CFI are used for control flow protections. Please explain them briefly, and describe one drawback of each. (4')

6. Xiao Mei wants to send his data to a untrusted cloud for computing. His computation only includes addition and multiplication operations. He wants to send encrypted data to the cloud and let the cloud do the computation without decryption. Which methods can he use to do so? Please briefly explain your reasons. (6')

我承诺，我将严格遵守考试纪律。

承诺人：_________

题号									
得分									
批阅人(流水阅卷教师签名处)									