ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Systems@ETH Zürich

Prof. T. Roscoe, Prof. R. Wattenhofer

Computer Systems

Quiz 3

Question 1

A shared coin...

a) ...is zero with probability 0.5 and one with probability 0.5

b) ...has to be generated by a trusted dealer

c) ...may fail with constant probability

d) ...cannot be implemented in an asynchronous setting if there are $f \geq 1$ crashes

Question 2

In practice, deterministic pseudo-random number generators (PRNGs) are often used together with some initialization parameter (seed). Do asynchronous byzantine agreement protocols still work if the adversary knows the PRNG and the seed?

a) Yes.

b) Yes, but they are slower due to imperfect randomness.

c) Only if the PRNG output looks random.

d) No.

Question 3

In Algorithm 19.15 (crash-resilient shared coin with blackboard), if $|C| \geq n^2$, then which of the following is correct?

a) All nodes have written at least n coins on the blackboard.

b) Each node wrote at least one coin (but not necessarily n coins) on the blackboard.

c) All coins on the blackboard might come from the same node.

d) All nodes are going to see the same sum.

Question 4

In Algorithm 19.15 (crash-resilient shared coin with blackboard) nodes first write and then read from the blackboard. What happens if we do it the other way around (first read the contents of the blackboard and check if you see enough coins to finish; if not, write on the blackboard)?

a) We need to increase the threshold from n^2 to n^3.

b) The algorithm no longer resists worst-case scheduling.

c) The algorithm tolerates fewer crashes.

d) (It still works.)

Question 5

In Algorithm 19.23 (threshold secret sharing), the dealer sends signed shares over authenticated channels. What could happen if the shares were not signed?

a) Only secret distribution may fail.

b) Only secret recovery may fail.

c) Both may fail.

d) (It still works.)

Question 6

Our final asynchronous byzantine agreement (ABA) protocol assumes a trusted dealer distributing the shares before the protocol starts. Let us now upgrade the dealer to a trusted-third-party (TTP). The TTP is honest, alive, and responsive for the whole duration of the protocol. What is true?

a) ABA becomes easy to solve: the TTP tells everybody who the byzantine parties are and they ignore them.

b) ABA becomes easy to solve: parties send their inputs to the TTP, which waits for n - f inputs and sends back the majority value for everyone to output.

c) ABA cannot be solved assuming a TTP.

d) ABA requires randomization and a complex protocol, even assuming a TTP.

Question 7

What would happen in Algorithm 19.8 (our first shared coin) if we removed lines 4-5 (nodes no longer broadcast coin sets and instead just decide based on their own set)? Assume that scheduling is random, i.e., every node receives a subset of coins of size $n - f$ uniformly at random.

a) The algorithm no longer meets the conditions of a shared coin.

b) The coin will output 0 too often.

c) The coin will output 1 too often.

d) (It still works.)

Question 8

Consider asynchronous byzantine agreement where each node has a **random** input 0 or 1 with equal probability. We want a protocol with constant failure probability. For what values of f is this possible with a **deterministic** protocol?

a) $f = 0$

b) $f < \frac{n}{10}$

c) $f < \frac{n}{3}$

d) $f < n$

Question 9

Algorithm 19.28 (synchronous byzantine shared coin with hashes and signatures) fails when the minimum hash value belongs to a byzantine node that distributed the signed round number only to some nodes (leading to conflicting answers). A proposed fix is to add an extra round of echoing: after getting the signatures in the first round, each node distributes all signatures it knows to everybody in a second round. Is this proposed fix correct?

a) Yes.

b) Yes, when $f \leq 1$.

c) Yes, when $f \leq \frac{n}{10}$.

d) No.

Question 10

A quorum system...

a) ...can consist of a single quorum

b) ...has work between 1 and n

c) ...has load between $\frac{1}{\sqrt{n}}$ and 1

d) (They are all true.)

Question 11

Concurrent Locking: assume quorums acquire locks in an arbitrary order (even adversarial). For how many concurrent quorums is this deadlock-free?

a) Not even for 1.

b) At most 1.

c) At most 2.

d) (It's always deadlock-free.)

Question 12

The Basic Grid quorum system...

a) ...is uniform

b) ...is minimal

c) ...can tolerate $c \cdot \sqrt{n}$ failures for some constant $c > 0$.

d) (They are all true.)

Question 13

B-Grid Quorum systems: In which scenario does a B-Grid Quorum system fail? (Note: This question has 2 right answers.)

a) The main diagonal fails.

b) In a single band, an element in each mini-column fails.

c) A whole column fails.

d) One element in each row fails.

Question 14

Quorum systems achieve mutual exclusion by requiring that any two quorums intersect in at least one element. If $\leq f$ locks are faulty and may be locked by multiple quorums simultaneously, mutual exclusion can be achieved if:

a) any two quorums intersect in at least 2 nodes

b) any two quorums intersect in at least f nodes

c) any two quorums intersect in at least $f + 1$ nodes

d) any two quorums intersect in at least $2f + 1$ nodes

Question 15

Consider a quorum system with 7 nodes where quorums consist of odd-size subsets with strictly more than half the nodes. How many quorums are there?