

程序代写代做 CS编程辅导



ECS855J/ECS796P
Distributed Systems
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

What ~~lecture~~ is about



Distributed consensus algorithms

- Introduction to consensus algorithms
- Paxos

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

What is that?



Distributed consensus algorithm with reaching agreement among a group of processes connected by an unreliable communications network.

WeChat: cstutorcs

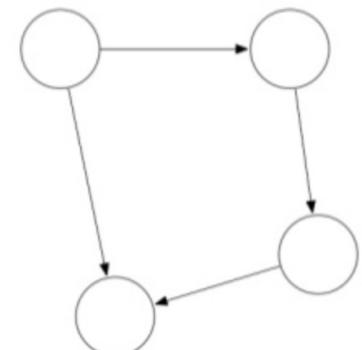
Assignment Project Exam Help

Email: tutorcs@163.com

"What do we eat for lunch?"

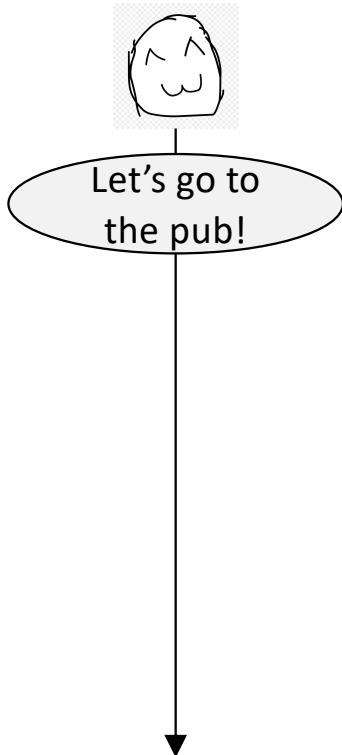
QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导

What is that?



WeChat: cstutorcs

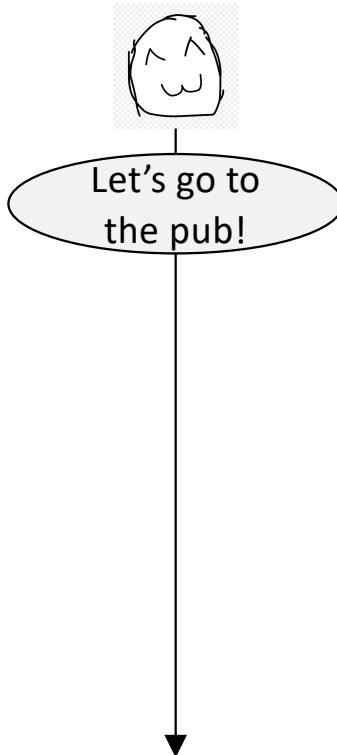
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导
What is that?



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

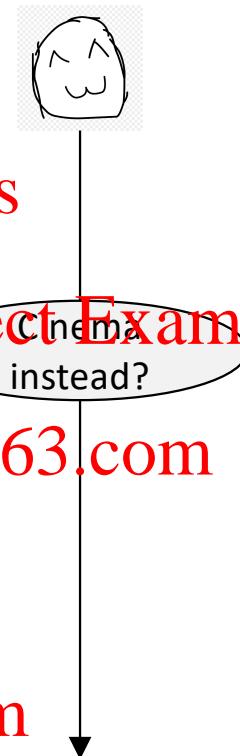
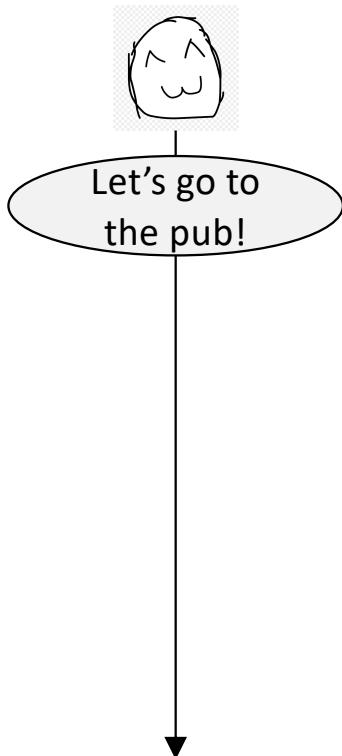
<https://tutorcs.com>

程序代写代做 CS编程辅导

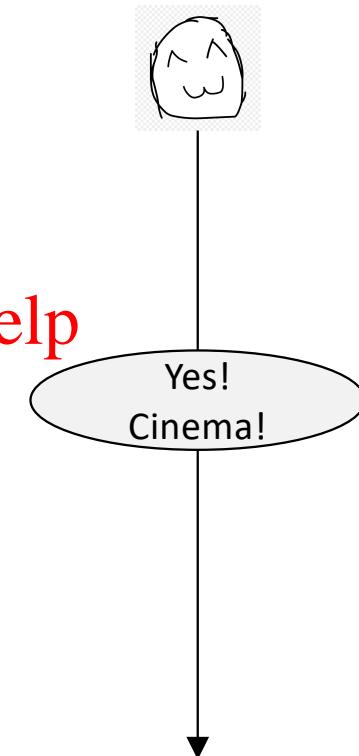
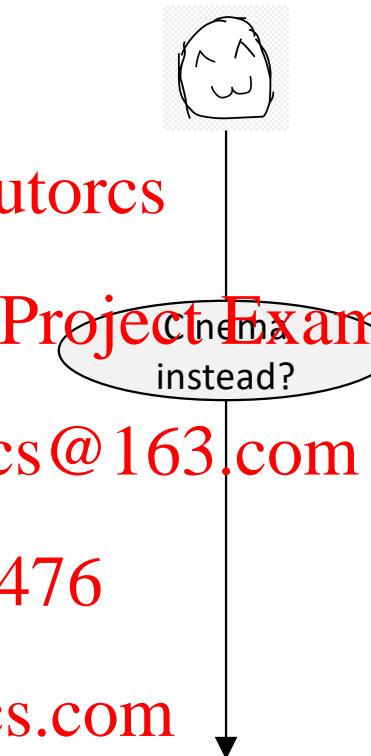
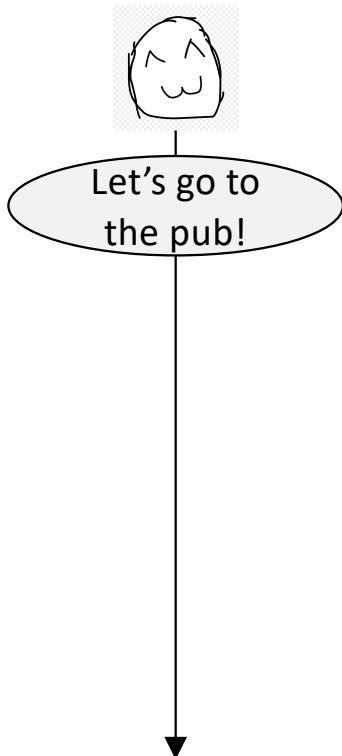
What is that?



Meanwhile, someone that message delayed, or was not listening...



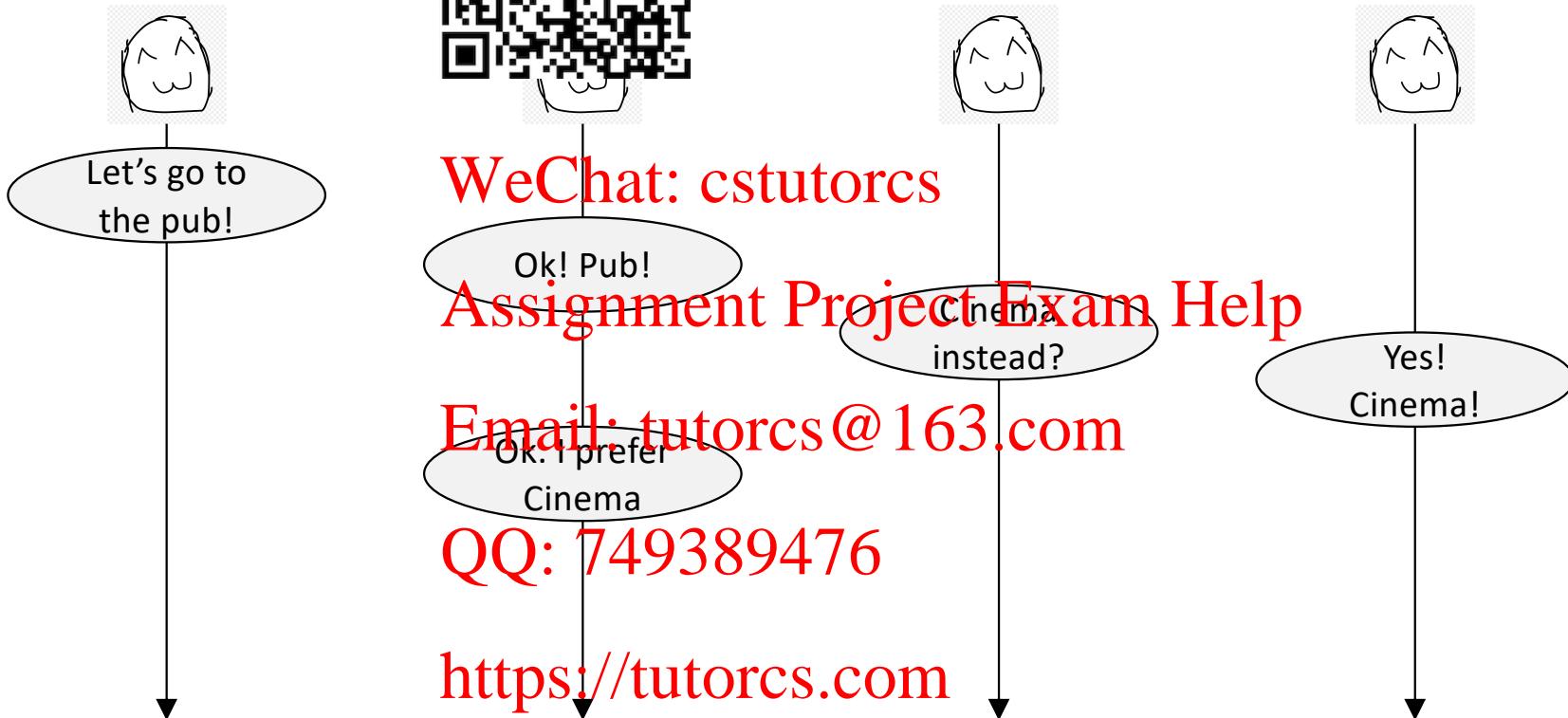
程序代写代做 CS编程辅导
What is that?



程序代写代做 CS编程辅导

What is that?

Someone needs to back off

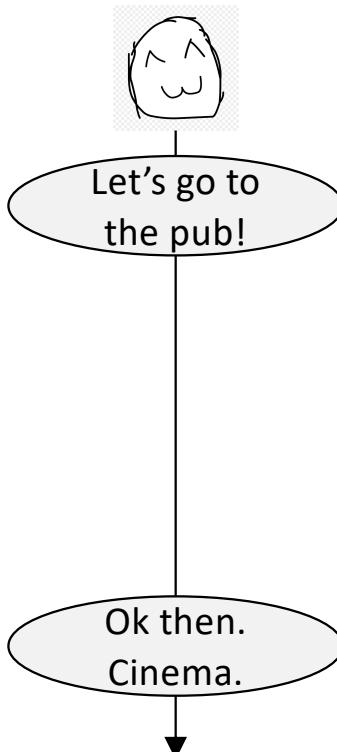


程序代写代做 CS编程辅导

What is that?



We are in democracy. Major



程序代写代做 CS编程辅导

So, what is consensus?



- Consensus is agreeing on it
- Once a majority agrees on a proposal, that is the consensus
- The reached consensus can be eventually known by everyone
 - Assignment Project Exam Help**
 - Email: tutorcs@163.com
 - QQ: 749389476
- The involved parties want to agree on any result, not on their proposal
 - Assignment Project Exam Help**
 - Email: tutorcs@163.com
 - QQ: 749389476
- Communication channels may be faulty, that is, messages can get lost
 - Assignment Project Exam Help**
 - Email: <https://tutorcs.com>

程序代写代做 CS编程辅导

Why do we care?



Example: You transfer money from one person to another. The money went out of one account but never arrived at the other one because a server or the network itself somewhere had failed.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导

Why do we care?



We need to be able to reliable agreement even though there are failures

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导

A bit of history



- 1985: FLP (Fisher-Lynch-Paterson) impossibility paper.

we cannot guarantee agreement in an asynchronous system where even one host might fail

WeChat: cstutorcs

Assignment Project Exam Help

The problem of consensus was known to be solvable in a synchronous setting, where processes could proceed in simultaneous steps

- The synchronous solution was resilient to faults: you can easily detect them!

Email: tutorcs@163.com
QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

A bit of history



- 1985: FLP (Fisher-Lynch-Paterson) impossibility paper.

we cannot guarantee agreement in an asynchronous system where even one host might fail

WeChat: cstutors

Assignment Project Exam Help

Asynchronous means:

- No upper bound on processing time
- No upper bound on clock drift rate
- No upper bound on networking delay

Email: tutores@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

A bit of history



- 1985: FLP (Fisher-Lynch-Pan执) impossibility paper.

we cannot guarantee agreement in an asynchronous system where even one host might fail.

WeChat: cstutorcs

Assignment Project Exam Help

WHY?

Email: tutorcs@163.com

We cannot detect reliably failures. We cannot know for sure the difference between a slow host/network and a failed host

<https://tutorcs.com>

程序代写代做 CS编程辅导

A bit of history



- 1985: FLP (Fisher-Lynch-Paterson) impossibility paper.

we cannot guarantee agreement in an asynchronous system where even one host might fail

WeChat: cstutors

Assignment Project Exam Help

The FLP result shows that in an asynchronous setting, where only one processor might crash, there is no distributed algorithm that solves the consensus problem

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

A bit of history – cont'd



- 1985: FLP impossibility proof
- 1989: Lamport: The Part-Time Parliament
WeChat: cstutorcs
- Distributed Systems are getting more important, thanks to Internet
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

How? (a bit of history)



- 1985: FLP impossibility proof
- 1989: Lamport: The Part-Time Parliament
WeChat: cstutorcs
- Distributed Systems are getting more important, thanks to Internet
Assignment Project Exam Help

Email: tutorcs@163.com

Let's remember the original WHY we could not guarantee agreement:

We cannot detect reliably failures.
QQ: 749389476

<https://tutorcs.com>

Can we categorize failures?

程序代写代做 CS编程辅导

Two types of failures



- Non-Byzantine
- Failed nodes stop communicating with other nodes
 - "Clean" failure
 - *Fail-stop* behavior
- Byzantine
- Failed nodes will keep sending messages
 - Incorrect and potentially misleading
 - Failed node becomes a *traitor*

WeChat: cstutorcs

Assignment Project Exam Help
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Two types of failures



- Non-Byzantine
- Failed nodes stop communicating with other nodes
 - "Clean" failure
 - *Fail-stop* behavior

- Byzantine
- Failed nodes will keep sending messages
 - Incorrect and potentially misleading
 - Failed node becomes a *traitor*

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

They are defined as arbitrary deviations of a process from its assumed behavior, e.g., software bug, a hardware malfunction, or a malicious attack.

<https://tutorcs.com>

程序代写代做 CS编程辅导

Two types of failures



- Non-Byzantine
- Failed nodes stop communicating with other nodes
 - "Clean" failure
 - *Fail-stop* behavior
- Byzantine
- Failed nodes will keep sending messages
 - Incorrect and potentially misleading
 - Failed node becomes a *traitor*

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

Assumption: https://tutorcs.com non-byzantine model

The goal for consensus



- We want agreement between processes (mutable states)
- Processes are concurrent, asynchronous and failure-prone

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Recap



- Consensus: making a decision (correctness) which is also correct (safety)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Liveness property



- Liveness: guarantee that **good** will happen

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Safety property



- Safety: guarantee that so-called bad will never happen
- Examples:
 - Real world: A peace treaty between nations provide safety as war will never happen
 - Consensus: no two processes decide on different value

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs
Paxos
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

Many slides from Ion Stoica presentation:

(<https://ucbrise.github.io/cs262a-spring2018/>)

<https://tutorcs.com>

程序代写代做 CS编程辅导

Paxos



- Paxos is a family of protocols for solving consensus in a network of unreliable processors

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Paxos



- Paxos is a family of protocols for solving consensus in a network of unreliable processors

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Paxos



- Paxos is a family of protocols for solving consensus in a network of unreliable processors

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Basic Paxos



- Leslie Lamport (one of the core developer of LaTeX!!!)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Basic Paxos



- Leslie Lamport (one of the core developer of LaTeX!!!)
- Deterministic and fault tolerant consensus protocol

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Basic Paxos



- Leslie Lamport (one of the core developer of LaTeX!!!)
- Deterministic and fault tolerant consensus protocol
- Named after a Greek Island

Assignment Project Exam Help

Email: tutorcs@163.com

(taken from the example Lamport
carries on his paper about elections
in the island)

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导

Basic Paxos



- Leslie Lamport (one of the core developer of LaTeX!!!)
- Deterministic and fault tolerant consensus protocol
- Named after a Greek Island
- Guarantees consistent results

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Does Paxos solve consensus?



- Provides safety and eventual consistency
- Safety:
 - Only a value which has been proposed can be chosen
 - Only a single value can be chosen
 - A process never learns a value unless it was chosen
- Eventual liveness:
 - If things go well, at some point in the future, consensus is eventually reached. However, this is not guaranteed.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Does Paxos solve consensus?



- FLP result still applies: Paxos guaranteed to reach consensus
- There is NO time bound
- We talk about eventual liveness

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

So simple, so obvious



"In fact, it is among the simplest and most obvious of distributed algorithms."

WeChat: cstutorcs

- Leslie Lamport

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Simple pseudocode



outcome[*p*] The decree written in *p*'s ledger, or BLANK if *p* has never voted yet.

lastTried[*p*] The number of the last ballot that *p* tried to begin, or none.

prevBal[*p*] The number of the last ballot in which *p* voted.

prevDec[*p*] The decree for which *p* last voted, or BLANK if *p* has never voted.

nextBal[*p*] The number of the last ballot in which *p* agreed to participate, or $-\infty$ if he has never agreed to participate in a ballot.

Next come variables representing information that priest *p* could keep on a slip of paper:

status[*p*] One of the following values:

idle Not conducting or trying to begin a ballot.

trying Trying to begin ballot number *lastTried*[*p*].

polling Now conducting ballot number *lastTried*[*p*].

If *p* has lost his slip of paper, then *status*[*p*] is assumed to equal *idle* and the values of the following four variables are irrelevant.

prevVotes[*p*] The set of votes received in *LastVote* messages for the current ballot (the one with ballot number *lastTried*[*p*]).

quorum[*p*] If *status*[*p*] = *polling*, then the set of priests forming the quorum of the current ballot; otherwise, meaningless.

voters[*p*] If *status*[*p*] = *polling*, then the set of quorum members from whom *p* has received *Voted* messages in the current ballot; otherwise, meaningless.

decree[*p*] If *status*[*p*] = *polling*, then the decree of the current ballot; otherwise, meaningless.

Try New Ballot

Always enabled.

- Set *lastTried*[*p*] to any ballot number *b*, greater than its previous value, such that *owner*(*b*) = *p*.
- Set *status*[*p*] to *trying*.
- Set *prevVotes*[*p*] to \emptyset .

Send NextBallot Message

Enabled whenever *status*[*p*] = *trying*.

- Send a *NextBallot*(*lastTried*[*p*]) message to any priest.

Receive NextBallot(*b*) Message

If *b* \geq *nextBal*[*p*] then

- Set *nextBal*[*p*] to *b*.

Send LastVote Message

Enabled whenever *nextBal*[*p*] > *prevBal*[*p*].

- Send a *LastVote*(*nextBal*[*p*], *v*) message to priest *owner*(*nextBal*[*p*]), where *v_{pst}* = *p*, *v_{bal}* = *prevBal*[*p*], and *v_{dec}* = *prevDec*[*p*].

Receive LastVote(*b*, *v*) Message

If *b* = *lastTried*[*p*] and *status*[*p*] = *trying*, then

- Set *prevVotes*[*p*] to the union of its original value and {*v*}.

Start Polling Majority Set *Q*

Enabled when *status*[*p*] = *trying* and *Q* \subseteq {*v_{pst}* : *v* \in *prevVotes*[*p*]}, where *Q* is a majority set.

- Set *status*[*p*] to *polling*.

- Set *quorum*[*p*] to *Q*.

- Set *voters*[*p*] to \emptyset .

- Set *decree*[*p*] to a decree *d* chosen as follows: Let *v* be the maximum element of *prevVotes*[*p*]. If *v_{bal}* \neq $-\infty$ then *d* = *v_{dec}*, else *d* can equal any decree.

Set *B* to the union of its former value and {*B*}, where *B_{dec}* = *d*, *B_{qrm}* = *Q*, *B_{vot}* = \emptyset , and *B_{bal}* = *lastTried*[*p*].

Send BeginBallot Message

Enabled when *status*[*p*] = *polling*.

- Send a *BeginBallot*(*lastTried*[*p*], *decree*[*p*]) message to any priest in *quorum*[*p*].

Receive BeginBallot(*b*, *d*) Message

If *b* = *nextBal*[*p*] and *p* is *Ballot* of the

- Set *prevBal*[*p*] to *b*.

- Set *prevDec*[*p*] to *d*.

- If there is a ballot *B* in *B* with *B_{bal}* = *b* [there will be], then choose any such *B* [there will be only one] and let the new value of *B* be obtained from its old value by setting *B_{vot}* equal to the union of its old value and {*p*}.

Send Voted Message

Enabled whenever *prevBal*[*p*] \neq $-\infty$.

- Send a *Voted*(*prevBal*[*p*], *p*) message to *owner*(*prevBal*[*p*]).

Receive Voted(*b*, *q*) Message

If *b* = *lastTried*[*p*] and *status*[*p*] = *polling*, then

- Set *voters*[*p*] to the union of its old value and {*q*}

Succeed

Enabled whenever *status*[*p*] = *polling*, *quorum*[*p*] \subseteq *voters*[*p*], and *outcome*[*p*] = BLANK.

- Set *outcome*[*p*] to *decree*[*p*].

Send Success Message

Enabled whenever *outcome*[*p*] \neq BLANK.

- Send a *Success*(*outcome*[*p*]) message to any priest.

Receive Success(*d*) Message

If *outcome*[*p*] = BLANK, then

- Set *outcome*[*p*] to *d*.

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com
QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

A political analogy

- A part-time parliament



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导 A political analogy



- Each round
 - Phase 1: A leader is elected (**election**)
 - Phase 2: Leader proposes a value (**propose**), processes acks
 - Phase 3: Leader multicast final value (**propose**)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导 Agents



- Three types of roles
 - Proposer: It receives a request from the client, and attempts to get a quorum of acceptors to agree upon it
WeChat: cstutorcs
 - Acceptor: It is a participant in the maintenance of the distributed storage. A state change in a Paxos cluster does not occur until a majority (quorum) of acceptors agree upon it
Assignment Project Exam Help
Email: tutorcs@163.com
 - Learner: It learns the agreed-upon value. They can be later queried to know what the consensus value was
QQ: 749389476
<https://tutorcs.com>

程序代写代做 CS编程辅导
In practice..



- Paxos nodes can take multiple roles even all of them
 - A single node can send proposals to other nodes, they can contribute to reaching consensus and ~~WeChat: cstutors~~ agreed upon value

Assignment Project Exam Help

- Paxos nodes must know how many acceptors a majority is
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导
In practice..



- Paxos nodes must be persistent: they cannot forget what they accepted
 - Even if the communication channel is faulty, they cannot forget
- A Paxos run aims at reaching a single consensus
 - Once a consensus is reached, it cannot progress to another consensus
 - In order to reach another consensus, a different Paxos run must happen

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Recall..



- Rounds are asynchronous
 - Time synchronization not required
 - If you are in round j and hear a message from round $j+1$, abort everything and move to round $j+1$
- WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

- Each round consists of three phases
 - Phase 1: A leader is elected (Election) QQ: 749389476
 - Phase 2: Leader proposes a value, processes acks (Bill)
 - Phase 3: Leader multicasts final value (Law)

Phase 1 – Election 程序代写代做 CS编程辅导

- Potential leader chooses a ballot ID, higher than anything it has seen so far
- Sends ballot ID to all processes
- Processes respond to higher ballot ID



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

Please elect me!

OK!

<https://tutorcs.com>

Phase 1 – Election 程序代写代做 CS编程辅导

- If majority (i.e., quorum) OK then you are the leader
 - If no one has majority go to next round



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

Please elect me!

OK!

<https://tutorcs.com>

Phase 2 – Proposal (BII)



- Leader sends proposal value v
 - Use $v=v'$ if some process already decided in a previous round and sent you its decided value v'
 - Otherwise propose its own value

WeChat: cstutorcs

Assignment Project Exam Help

- Recipient log on disk, and responds OK

Email: tutorcs@163.com

Value QQ: 749389476

Please elect me!

OK!

OK!

<https://tutorcs.com>

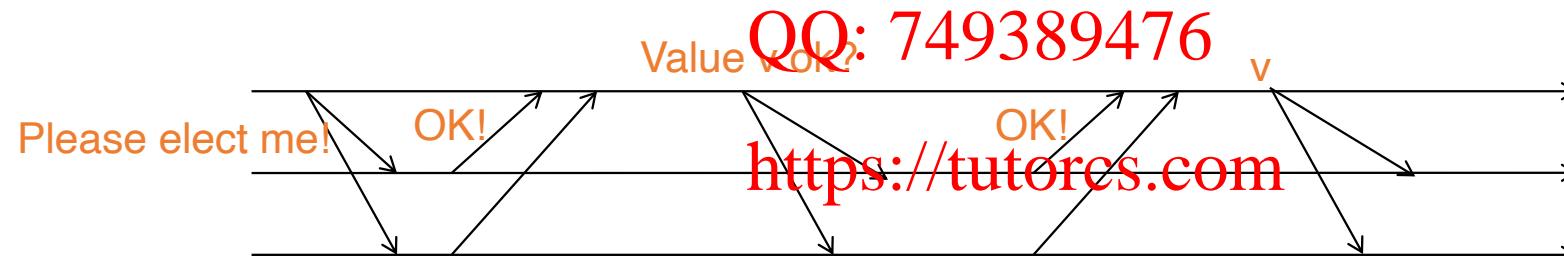
Phase 3 – Decision (Law) 程序代写代做 CS 编程辅导



- If leader hears OKs from recipients, it lets everyone know of the decision
- Recipients receive decisions, log it on disk
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com



程序代写代做 CS编程辅导

When is Consent Achieved?

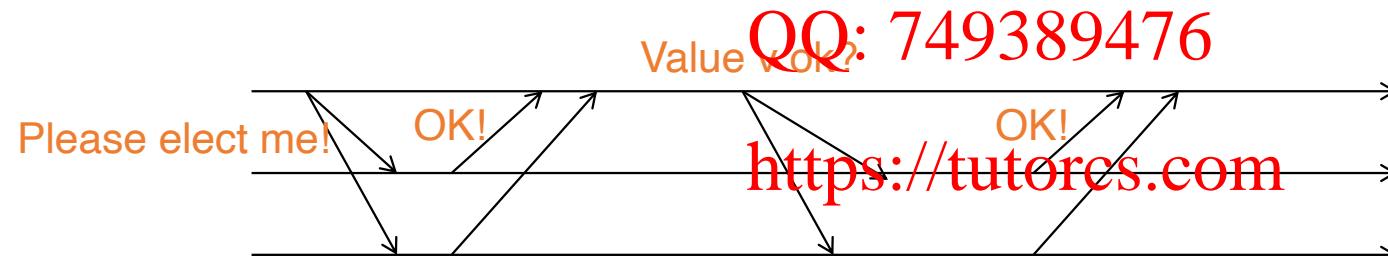


WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Value QQ: 749389476



程序代写代做 CS编程辅导

When is Consensus Achieved?

- When a majority of processes proposed value and accept it:



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Value: QQ: 749389476

Please elect me!

OK!

OK!

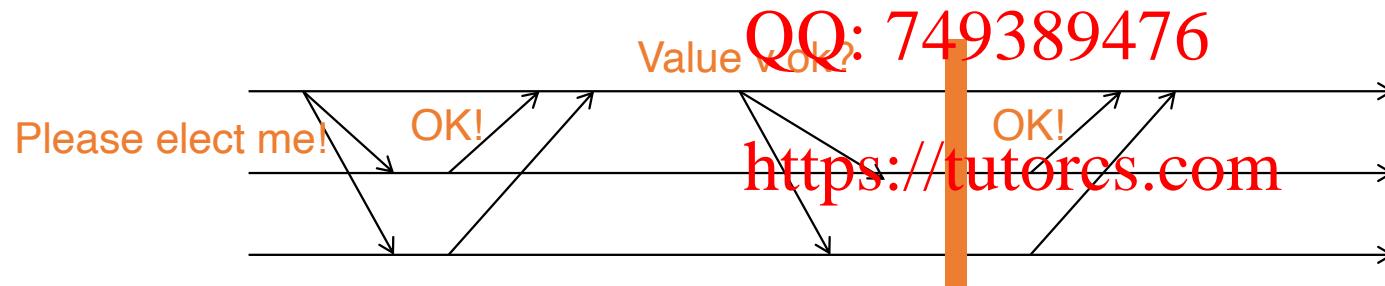
<https://tutorcs.com>

程序代写代做 CS编程辅导

When is Consensus Achieved?



- When a majority of processes have proposed value and accept it:
 - Are about to respond (about to respond) with OK!
- At this point decision has been made even though
 - Processes or even leader may not know!
- What if leader fails after that?



程序代写代做 CS编程辅导

Easy right? ☺



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Value QQ: 749389476

Please elect me!

OK!

<https://tutorcs.com>

程序代写代做 CS编程辅导

Easy right? ☺



- Let's have a look in more details now...

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Value QQ: 749389476

Please elect me!

OK!

<https://tutorcs.com>

OK!

Basic Paxos Protocol

Phase 1a: “Prepare”

Select proposal number* N and send a $\text{Prepare}(N)$ request to a quorum of acceptors.



Proposer

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

* = record to stable storage

Basic Paxos Protocol

Phase 1a: "Prepare"

Select proposal number* N and send a $\text{Prepare}(N)$ request to a quorum of acceptors.



Proposer

Phase 1b: "Promise"

If $N > \text{number of accepted proposals}$, promise to never accept any future proposal less than N ,

- * promise to never accept any future proposal less than N ,
- send a $\text{promise}(N, U)$ response

(where U is the highest-numbered proposal accepted so far (if any))

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Acceptor

* = record to stable storage

Basic Paxos Protocol

Phase 1a: "Prepare"

Select proposal number* N and send a $\text{propose}(N)$ request to a quorum of acceptors.



Proposer

Phase 1b: "Promise"

If $N > \text{number of previous promises or acceptances}$,

- * promise to never accept any future proposal less than N ,
- send a $\text{promise}(N, U)$ response

(where U is the highest-numbered proposal accepted so far (if any))

Phase 2a: "Accept!"

If proposer received promise responses from a quorum,

- send an $\text{accept}(N, W)$ request to those acceptors

(where W is the value of the highest-numbered proposal among the promise responses, or any value if no promise contained a proposal)

QQ: 749389476

<https://tutorcs.com>

Acceptor

* = record to stable storage

Basic Paxos Protocol

Phase 1a: "Prepare"

Select proposal number* N and send a $\text{propose}(N)$ request to a quorum of acceptors.



Proposer

Phase 1b: "Promise"

If $N > \text{number of any previous promise}$,

- * promise to never accept any future proposal less than N ,
- send a $\text{promise}(N, U)$ response

(where U is the highest-numbered proposal accepted so far (if any))

Phase 2a: "Accept!"

Assignment Project Exam Help

If proposer received promise responses from a quorum,

- send an $\text{accept}(N, W)$ request to those acceptors

(where W is the value of the highest-numbered proposal among the promise responses, or any value if no promise contained a proposal)

Acceptor

Phase 2b: "Accepted"

If $N \geq \text{number of any previous promise}$,

- * accept the proposal
- send an accepted notification to the learner

* = record to stable storage

程序代写代做 CS编程辅导

Milestones



- If the majority of acceptors accept (ID_p , v), no $ID < ID_p$ can make it through
- If a majority of acceptors accept $(ID_p, value)$, consensus is reached. Consensus is and will always be on a value

WeChat: cstutorcs

Email: tutorcs@163.com

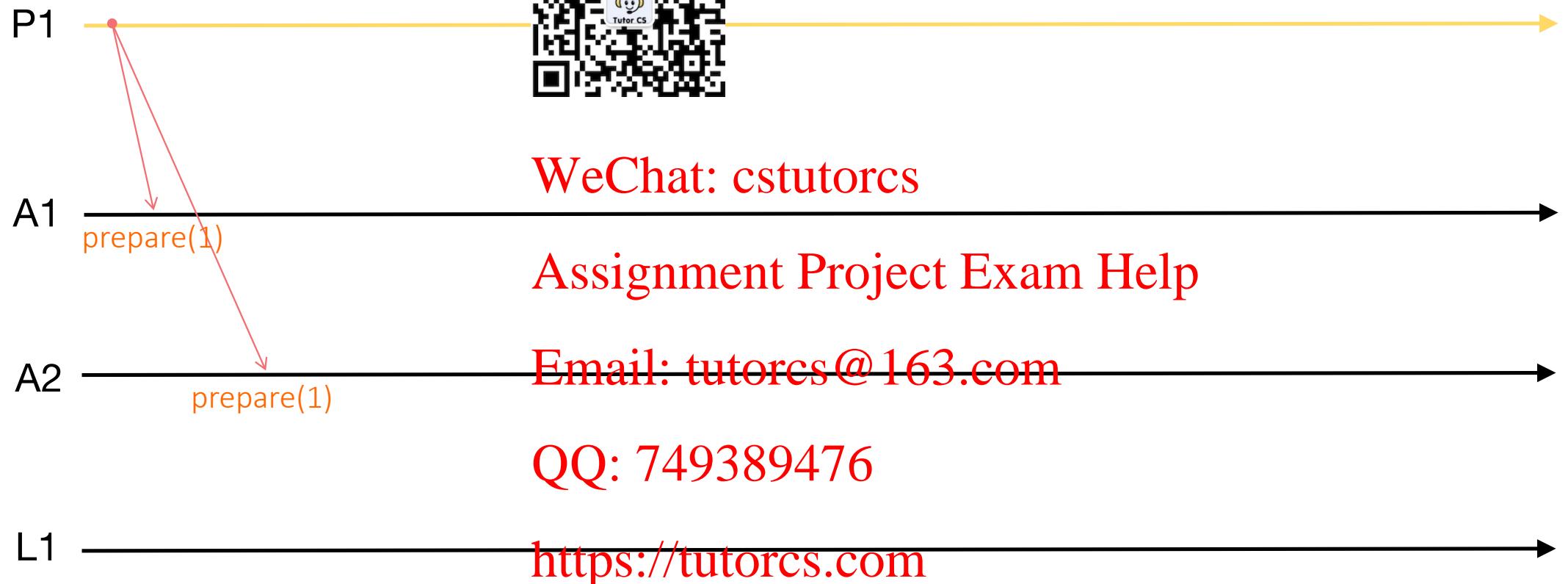
- If a proposer/learner gets the majority of accept for a specific ID_p , they know that consensus has been reached on a value

QQ: 749389476

<https://tutorcs.com>

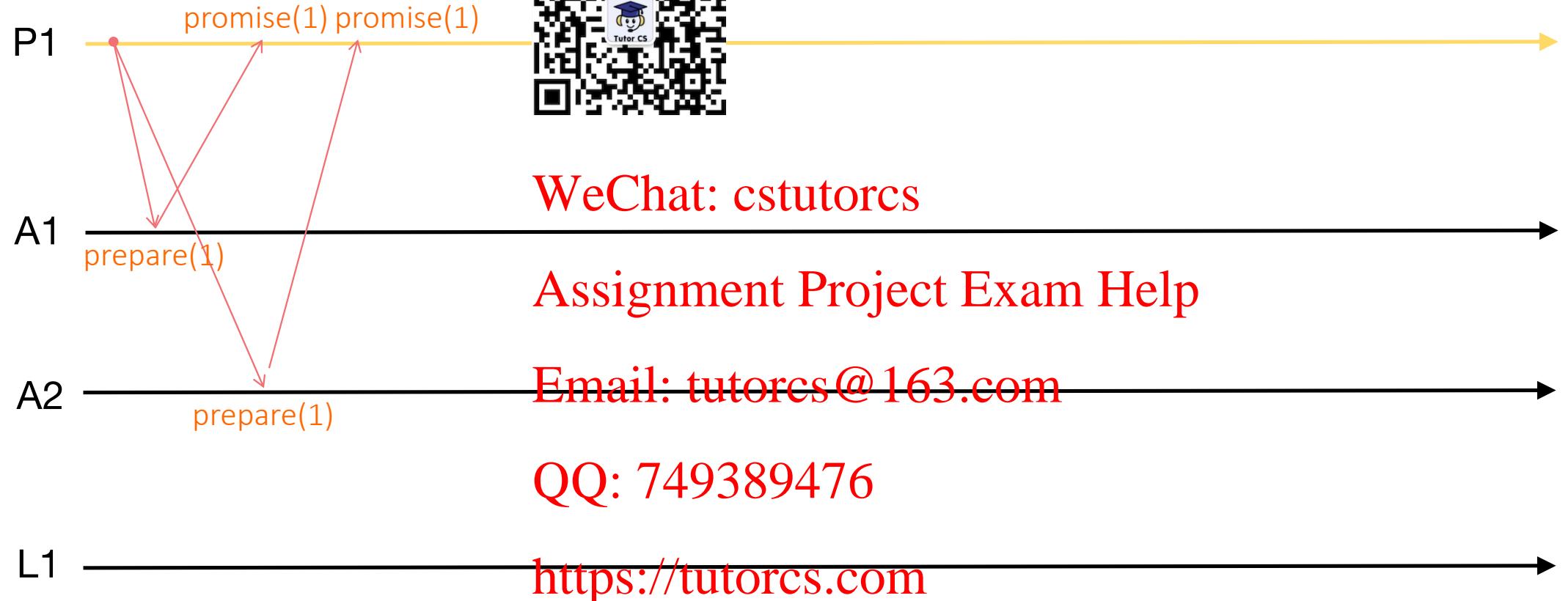
程序代写代做 CS编程辅导

Time 0: P1 waits for proposal “A”



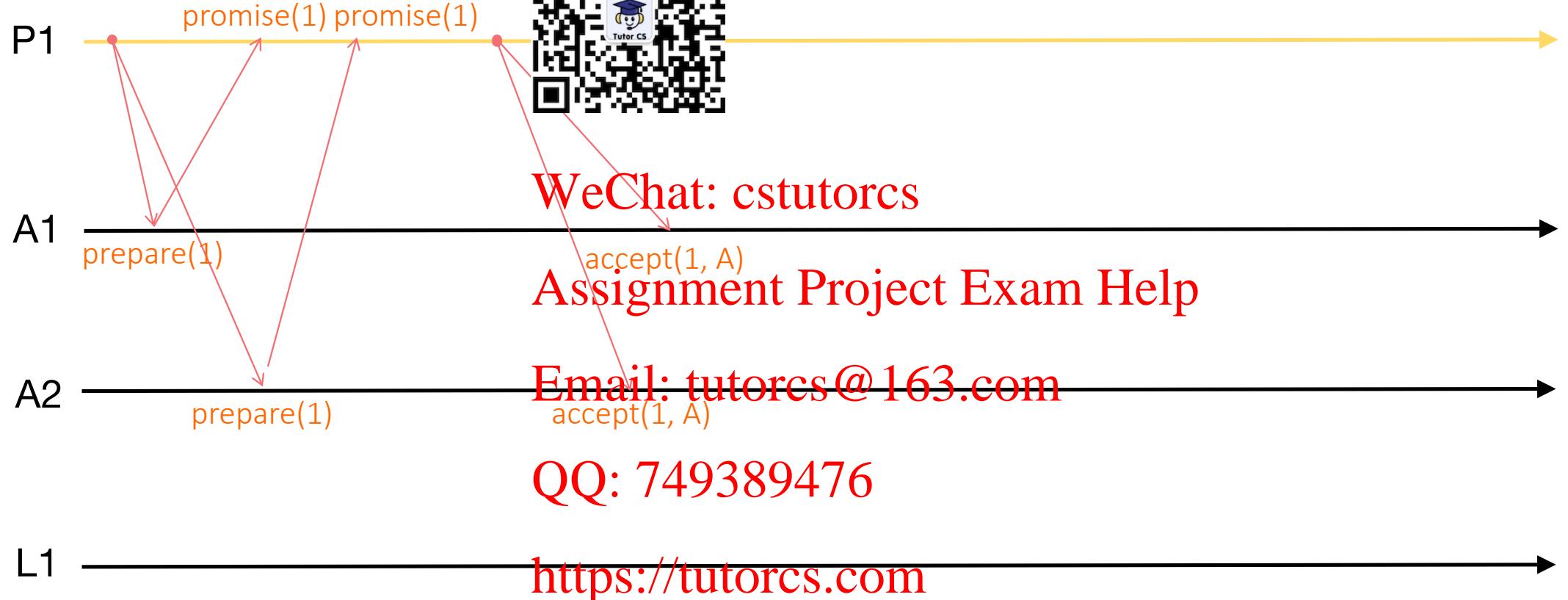
程序代写代做 CS编程辅导

Time 0: P1 waits for proposal “A”



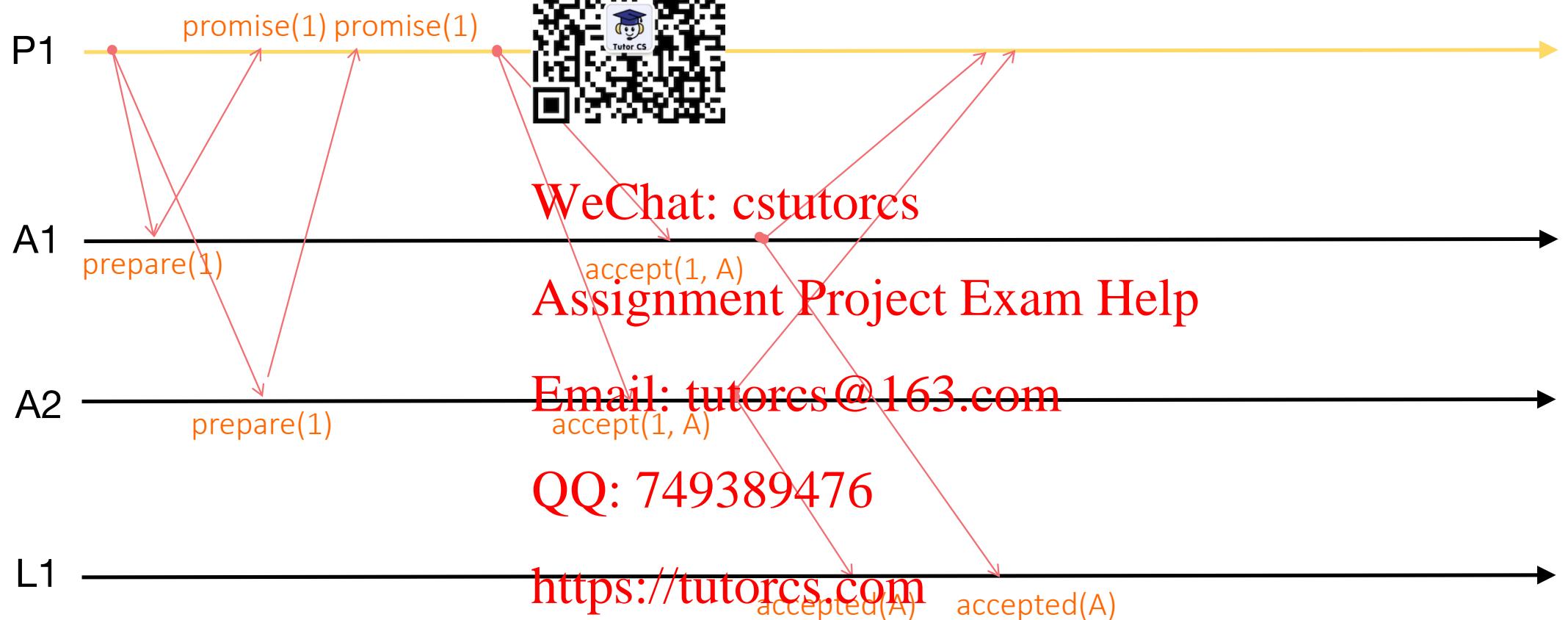
程序代写代做 CS编程辅导

Time 0: P1 waits for proposal “A”



程序代写代做 CS编程辅导

Time 0: P1 waits for proposal “A”



程序代写代做 CS编程辅导
If the acceptor has accepted something before..



<https://tutorcs.com>

程序代写代做 CS编程辅导

It needs to reply with PROMISE ID and (accepted ID, value)

P1



(10, (5, A))

A1

Highest Accept: (5, A)

Highest Prepare: 15

A2

Highest Accept: (5, A)

Highest Prepare: 8

prepare(10)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

prepare(10) Highest Accept: (5, A)

QQ: 749389476

Highest Prepare: 10

<https://tutorcs.com>

程序代写代做 CS编程辅导

The proposer needs to pick the value with the highest ID that

P1

Assume it got
promise for 10
from a quorum



A1

Highest Accept: (5, A)
Highest Prepare: 15

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

accept(10, A)

QQ: 749389476

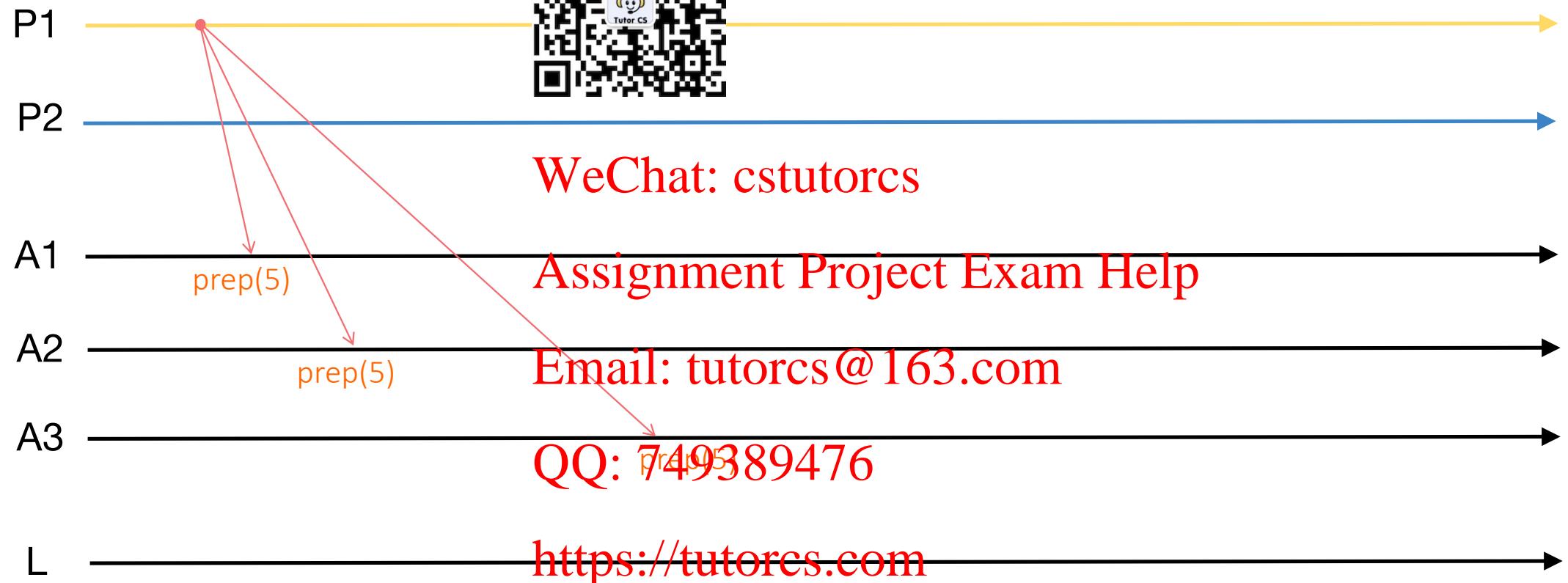
:

L1 https://tutorcs.com

accept(10, A)

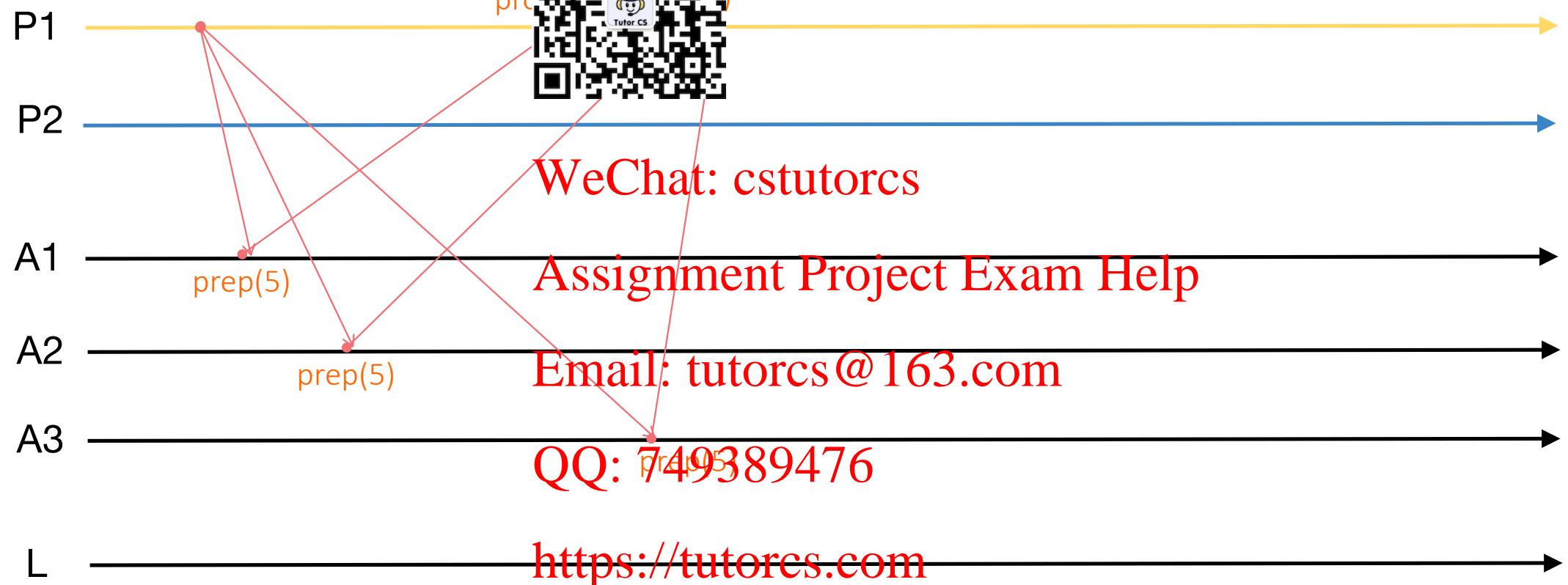
程序代写代做 CS编程辅导

Recap example want to propose value A



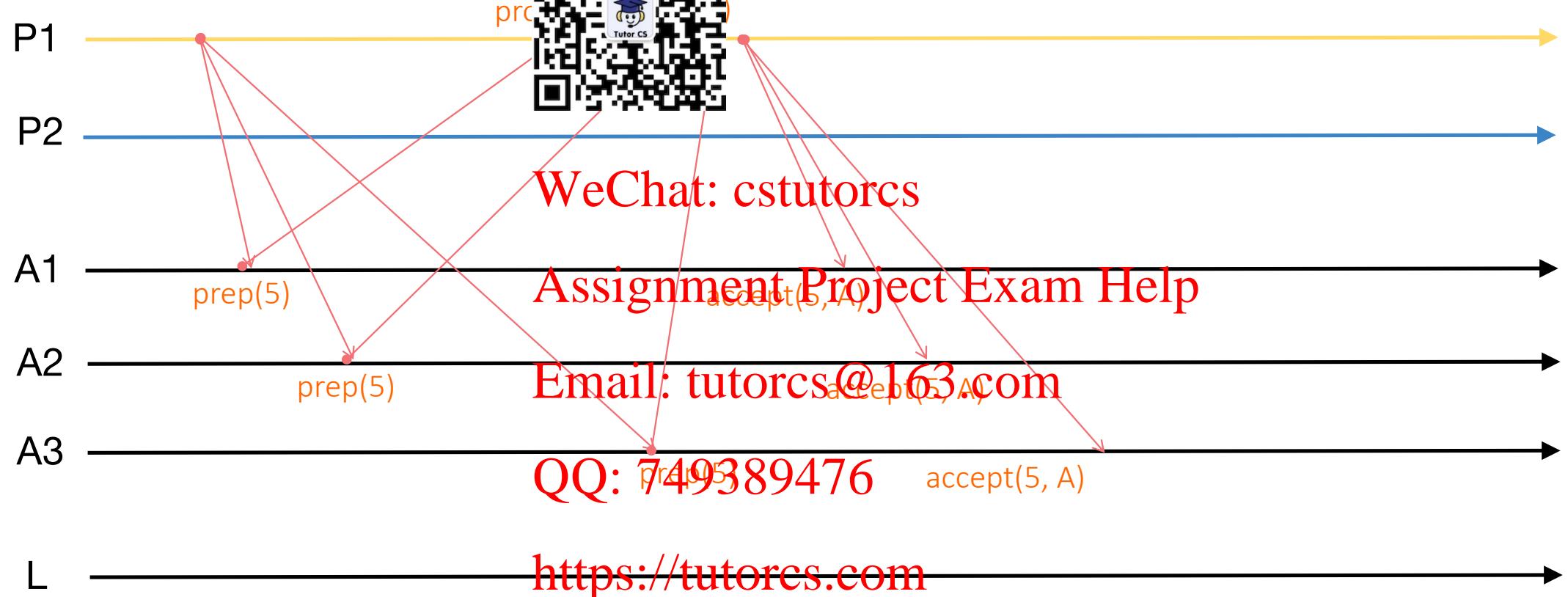
程序代写代做 CS编程辅导

Recap example want to propose value A



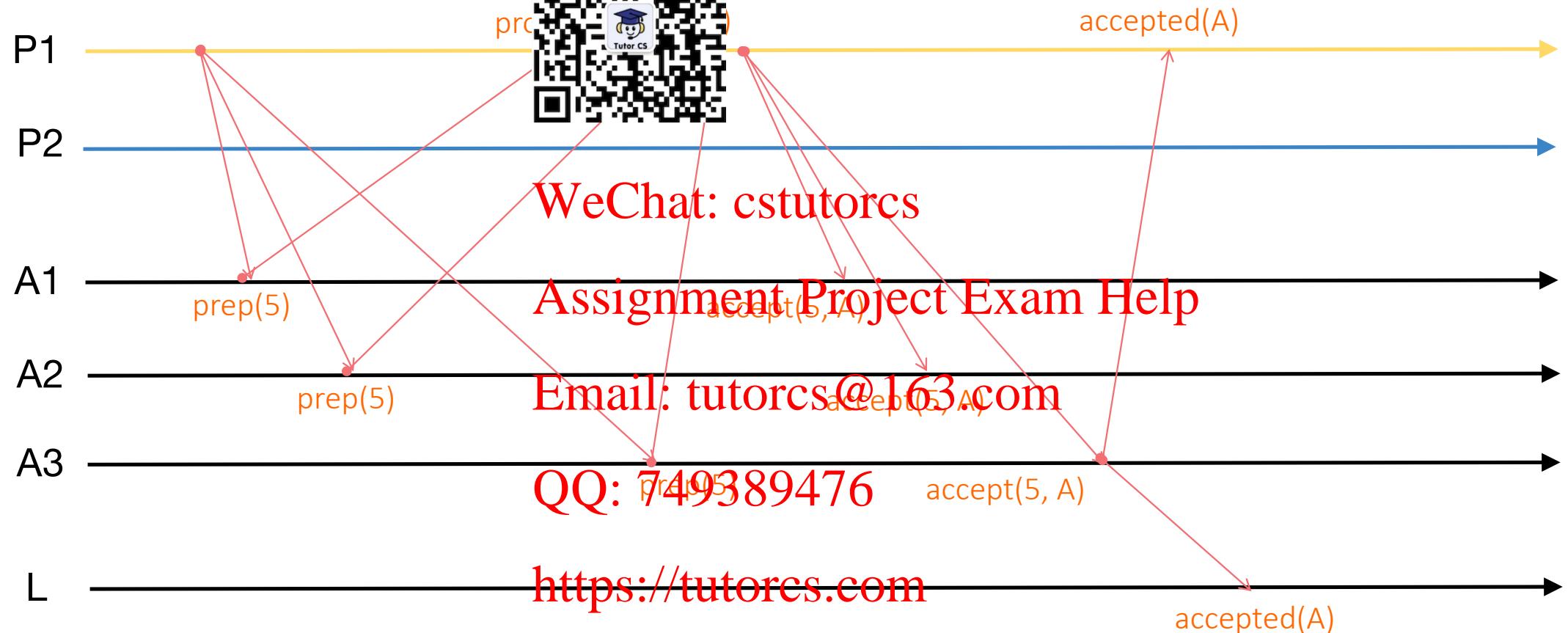
程序代写代做 CS编程辅导

Recap example want to propose value A



程序代写代做 CS编程辅导

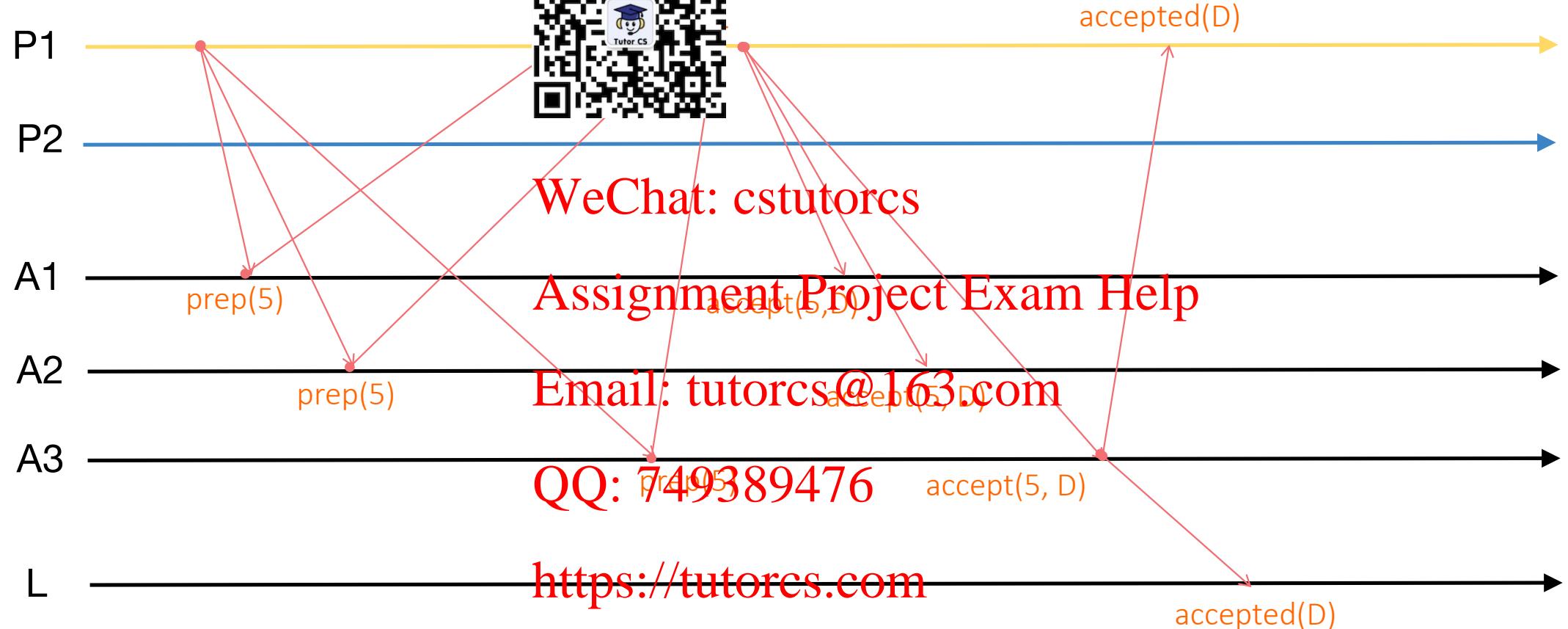
Recap example want to propose value A



程序代写代做 CS 编程辅导



Recap Example



程序代写代做 CS编程辅导

What can go wrong (liveness)



- Process fails
 - still works as long as many up
- Leader fails
 - Start another round

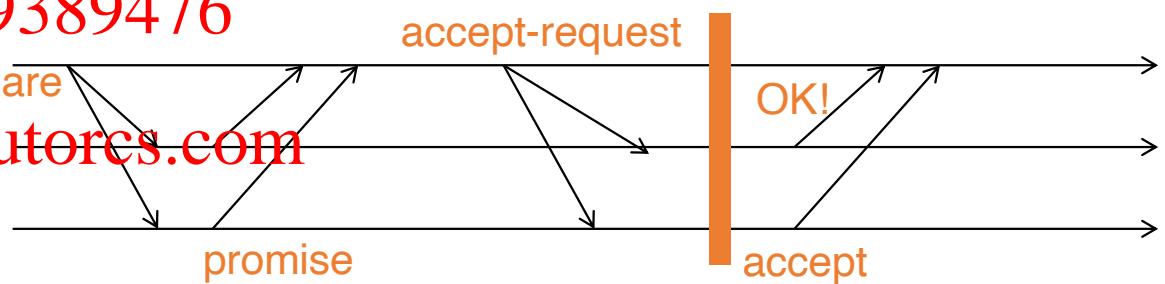
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导

What can go wrong (liveness)



- Message dropped
 - If too flaky, start another round
- Note that anyone can start a round any time

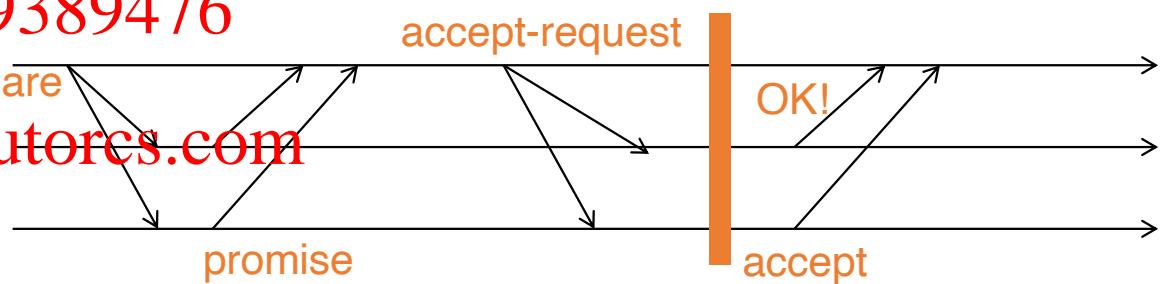
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导

What can go wrong (liveness)



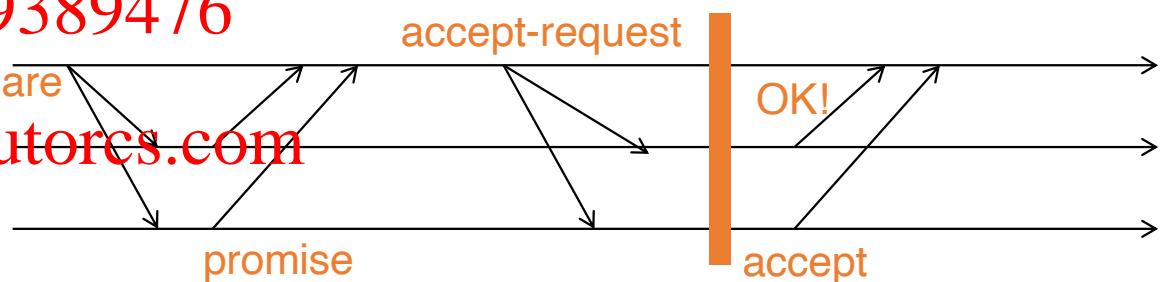
- Protocol may never end – deadlock, buddy!
 - Impossibility result not provable
 - If things go well sometime in the future, consensus reached!
- Example: two or more simultaneous proposer

WeChat: cstutorcs

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导

Livelock



P1

WeChat: cstutorcs

A1

Assignment Project Exam Help

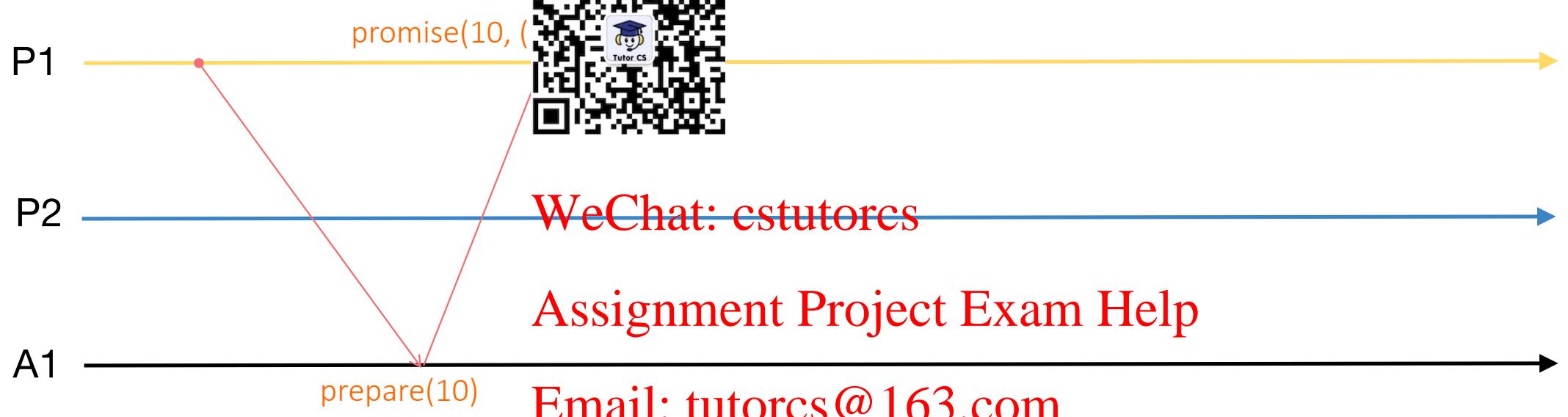
Email: tutorcs@163.com

QQ: 749389476
A1: Highest accept; (5, A)

Highest prepare: 8
<https://tutorcs.com>

程序代写代做 CS编程辅导

Livelock

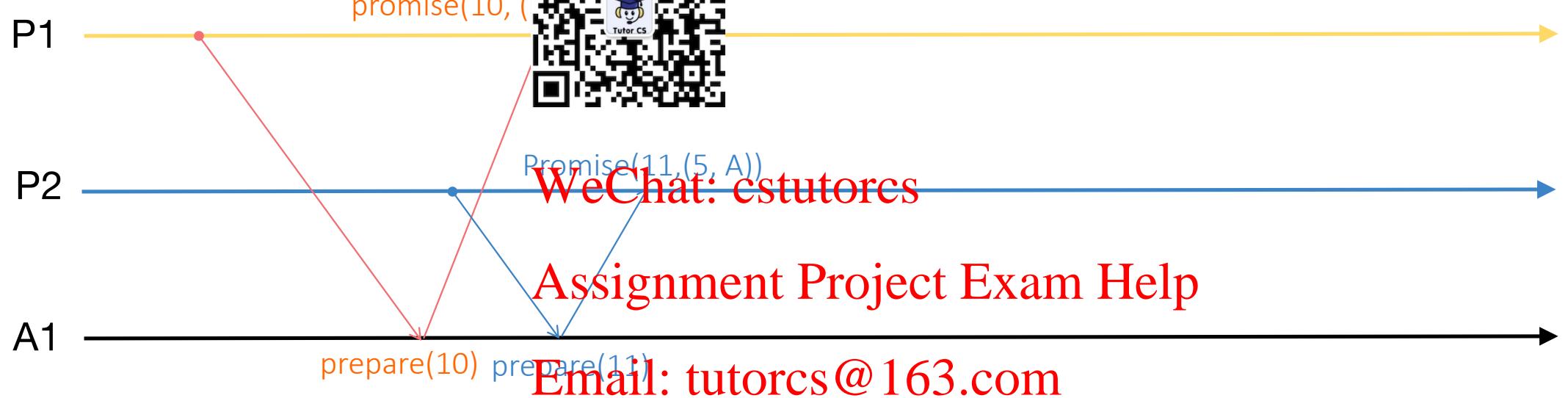


QQ: 749389476
A1: Highest accept; (5, A)

Highest prepare: 10
<https://tutorcs.com>

程序代写代做 CS编程辅导

Livelock

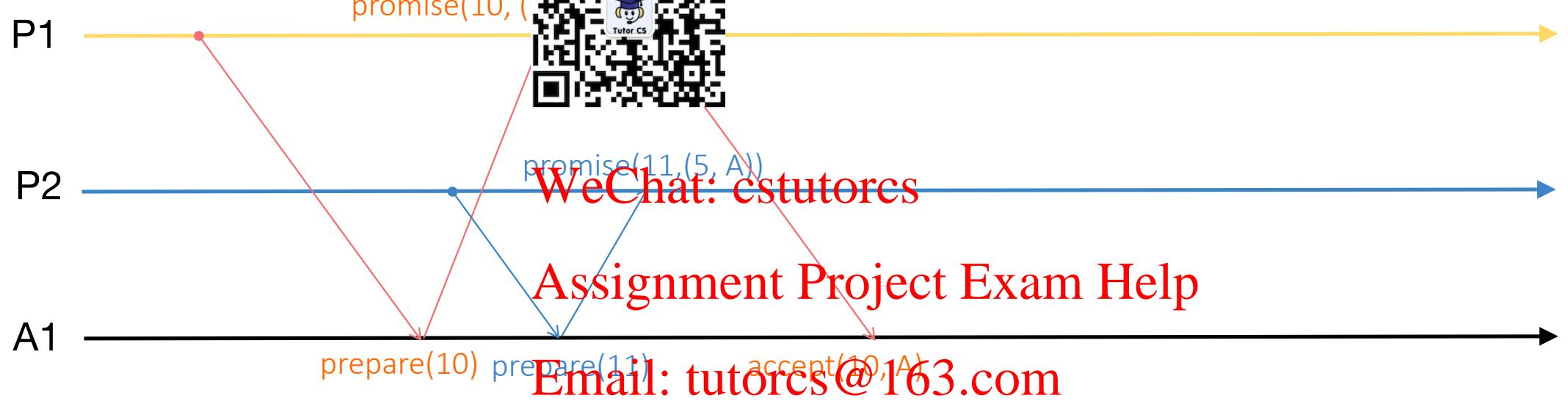


QQ: 749389476
A1: Highest accept; (5, A)

Highest prepare: 11
<https://tutorcs.com>

程序代写代做 CS编程辅导

Livelock



程序代写代做 CS 编程辅导

Livelock



QQ: 749389476
A1: Highest accept; (5, A)

Highest prepare: 12
<https://tutorcs.com>

程序代写代做 CS 编程辅导

Livelock



QQ: 749389476
A1: Highest accept; (5, A)

Highest prepare: 13
<https://tutorcs.com>

程序代写代做 CS 编程辅导

Livelock



QQ: 749389476

So now?

<https://tutorcs.com>

程序代写代做 CS编程辅导

Livelock



QQ: 749389476

This is a hot spot that can stall the Paxos run. A solution is to set an exponential back off in place.
<https://tutorcs.com>

程序代写代做 CS 编程辅导

Livelock



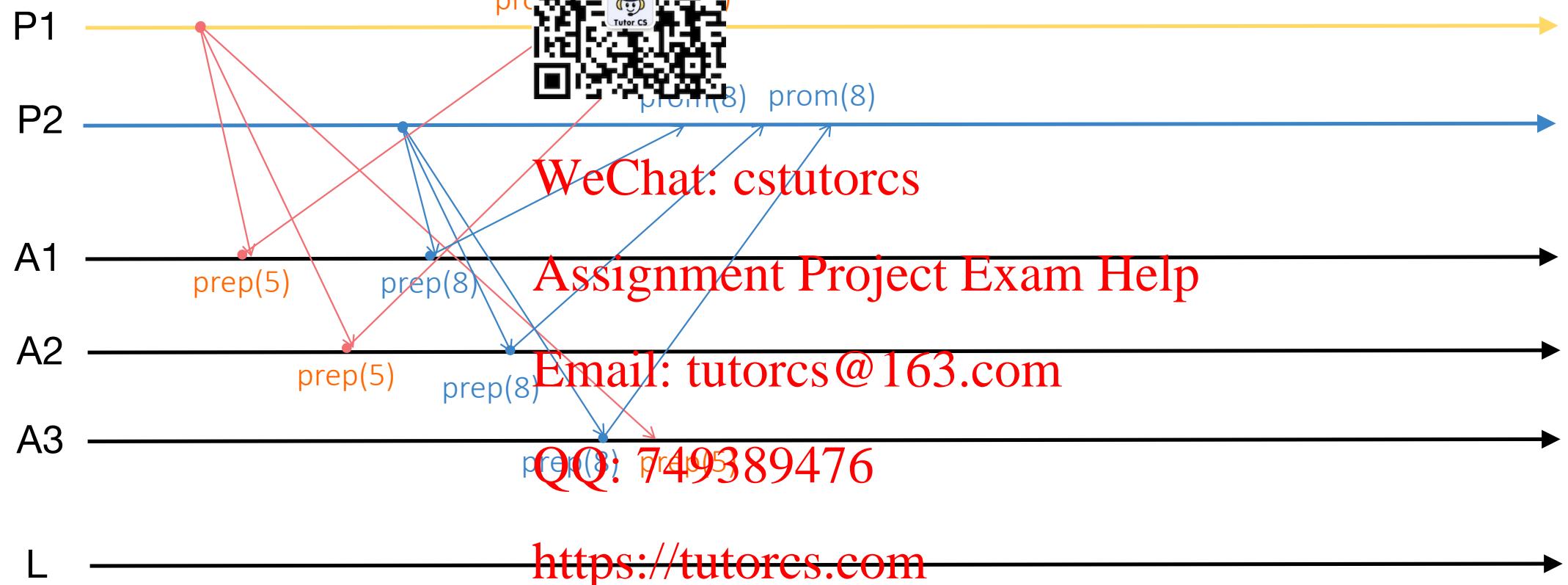
QQ: 749389476

Indeed, if there is enough time....

<https://tutorcs.com>

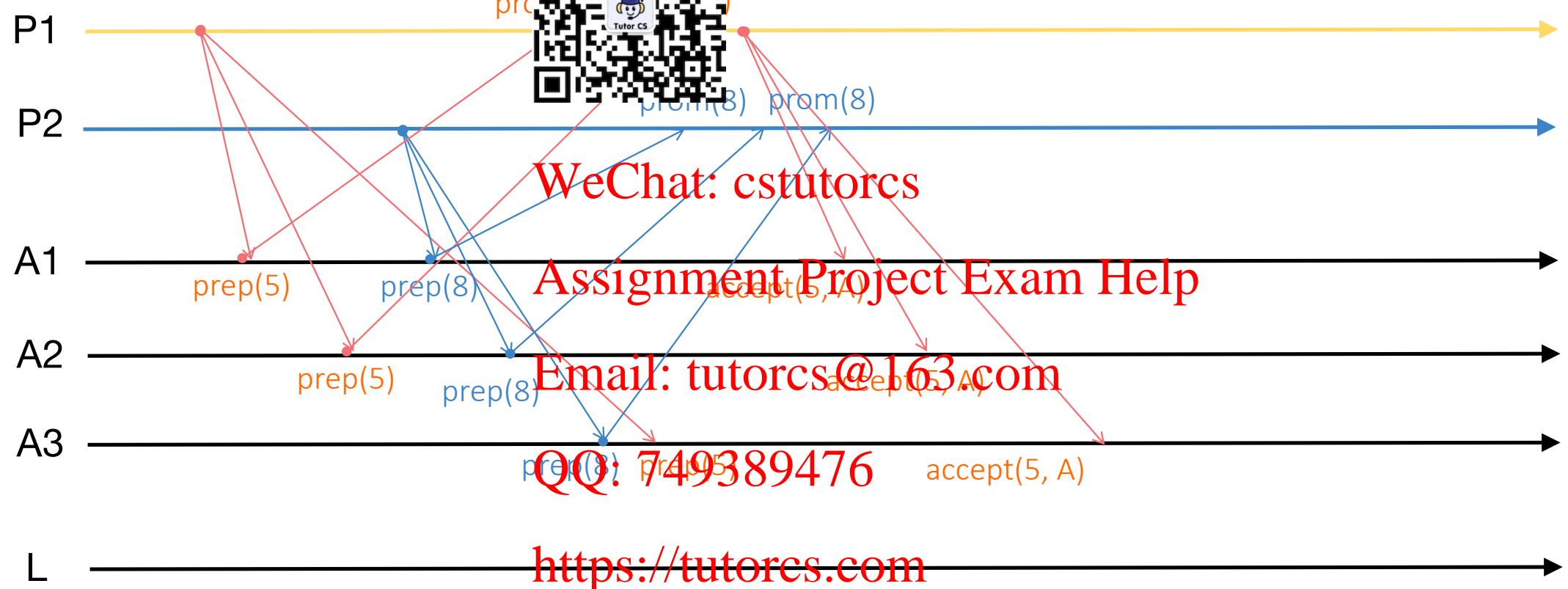
程序代写代做 CS编程辅导

P1 wants A, and P2 wants B



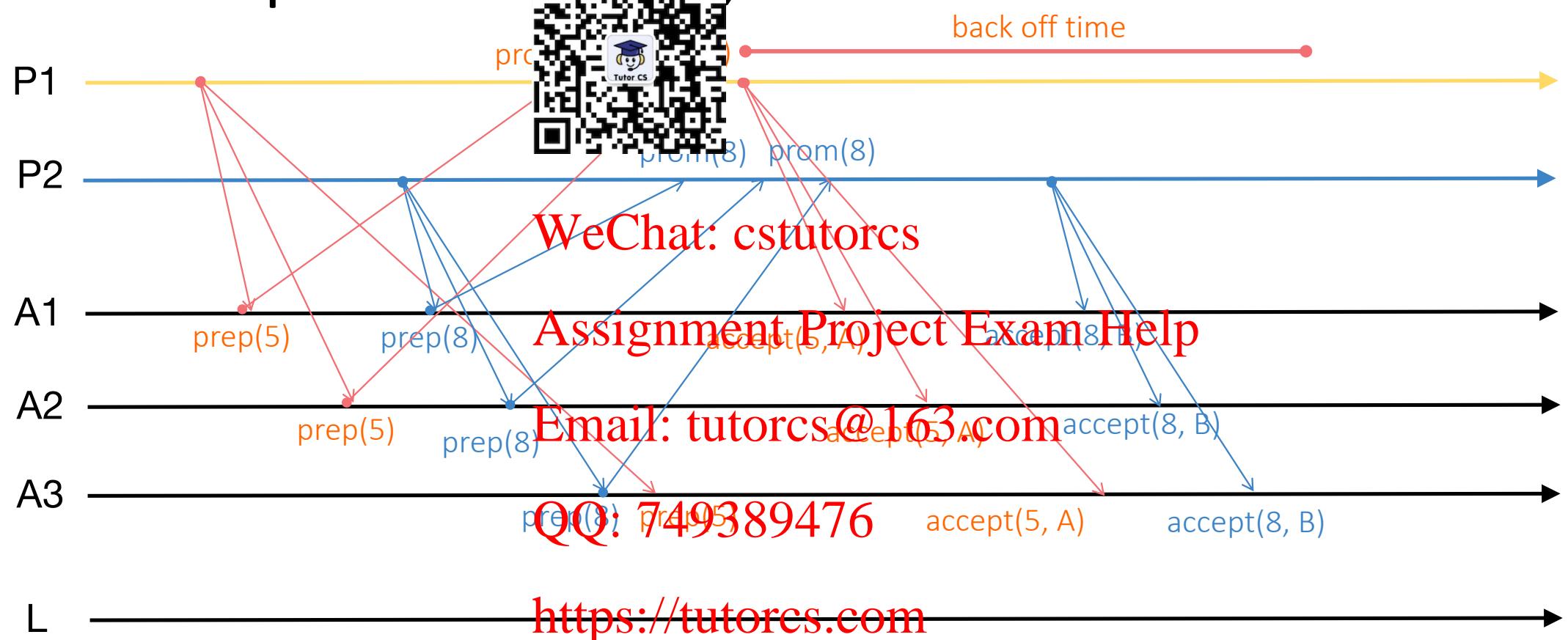
程序代写代做 CS编程辅导

Example: P1 wants A, and P2 wants B



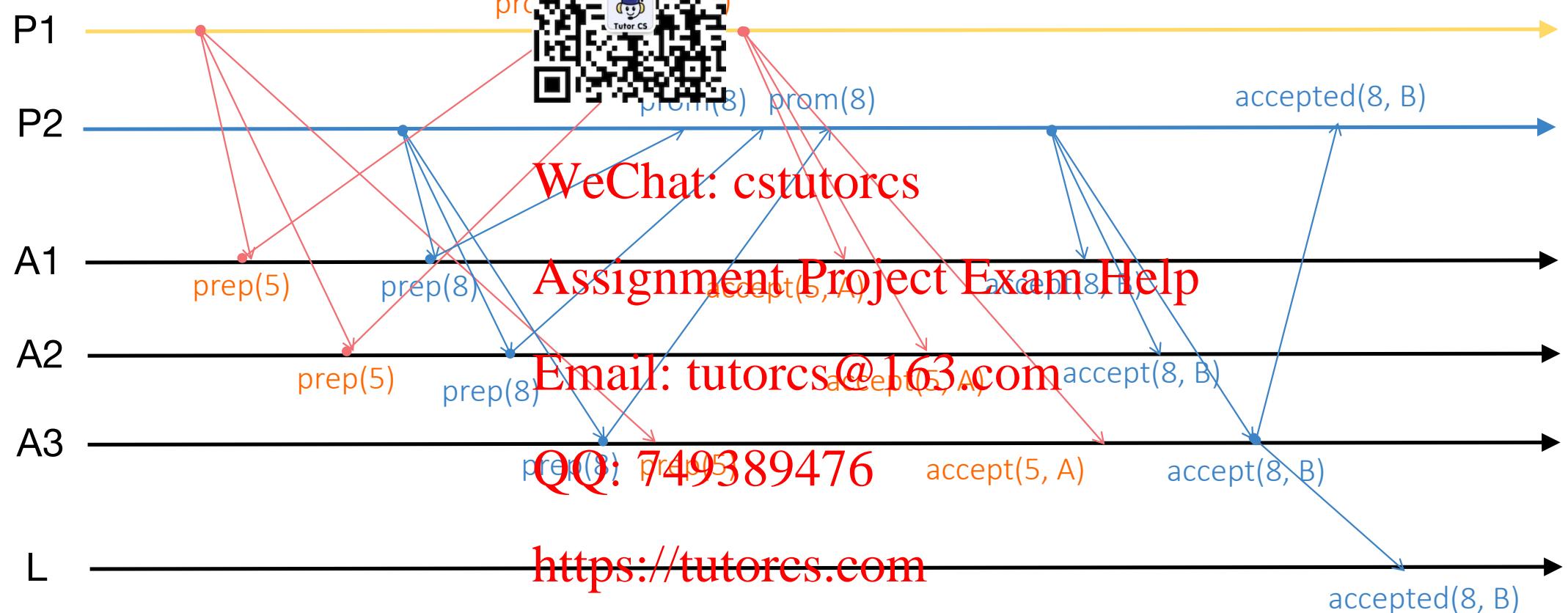
程序代写代做 CS编程辅导

Example: P1 wants A, and P2 wants B



程序代写代做 CS编程辅导

Example: P1 wants A, and P2 wants B



程序代写代做 CS编程辅导

Others



- Acceptors might send “N~~A~~” responses if they are not going to accept a proposal. This would tell the proposer that it can stop its attempt to create consensus with proposal N

Assignment Project Exam Help

- We said the proposal number needs to be strictly increasing and globally unique. How to do it? Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

Others



- Acceptors might send “NACK” responses if they are not going to accept a proposal. This would tell the proposer that it can stop its attempt to create consensus with proposal N

Assignment Project Exam Help

- We said the proposal number needs to be strictly increasing and globally unique. How to do it? Email: tutorcs@163.com

Tick: set low-order bits to proposer's (server) ID
QQ: 749389476

<https://tutorcs.com>