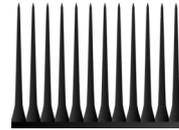
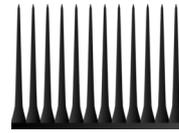


Peterson's algorithm oversimplified



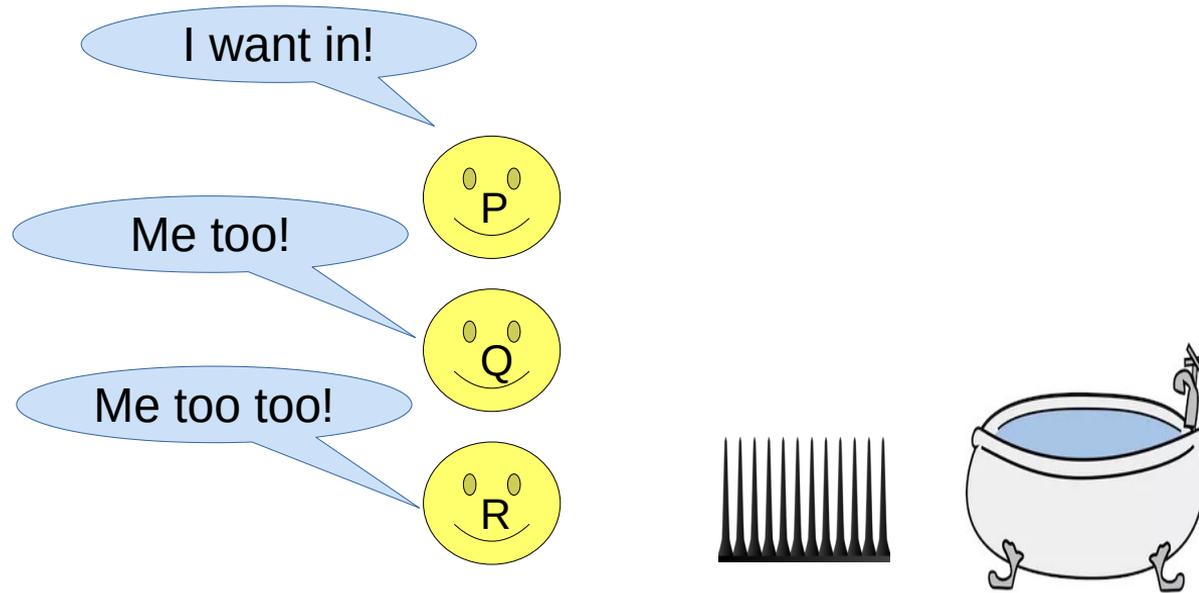
Peterson's algorithm oversimplified



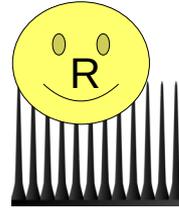
Critical section

Spike trap holds one person.
If >2 people try to pass,
last one in gets caught

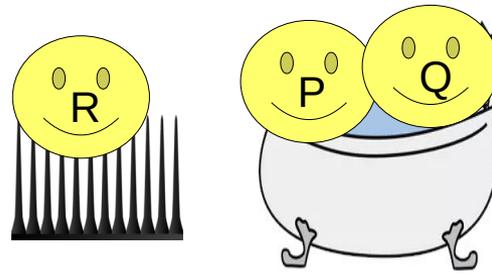
Peterson's algorithm oversimplified



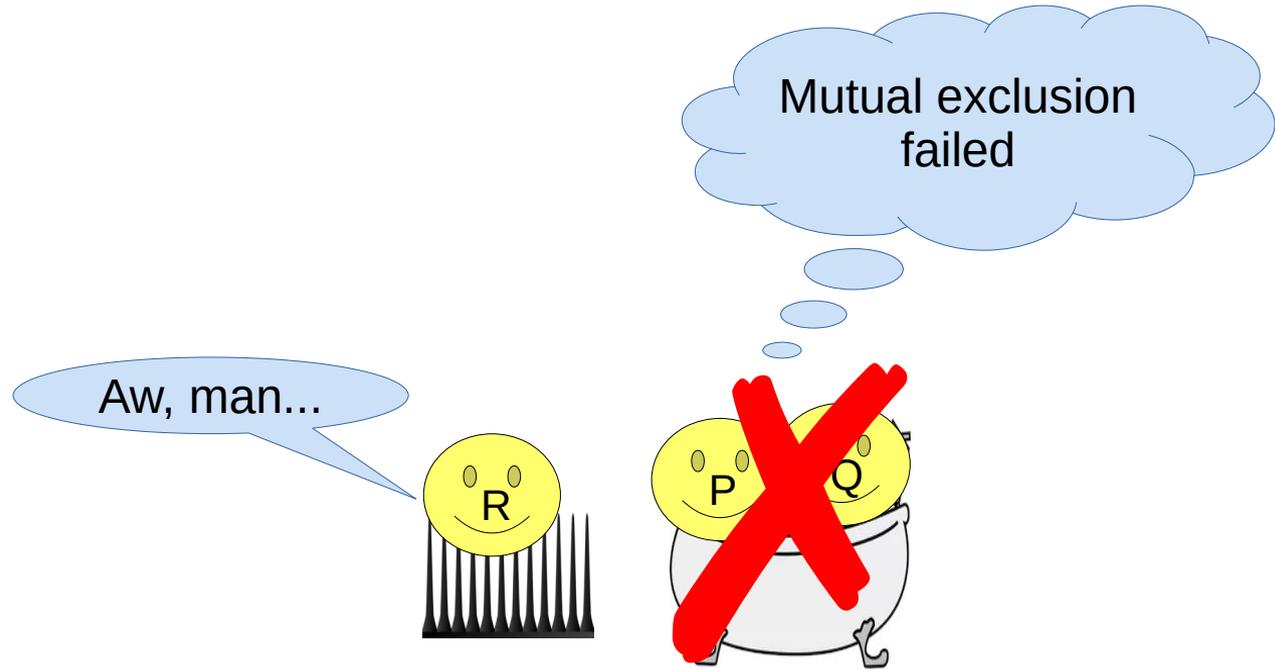
Peterson's algorithm oversimplified



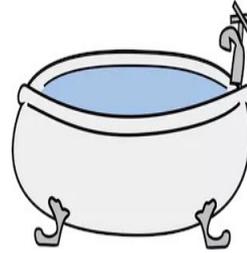
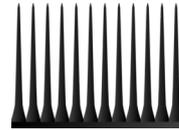
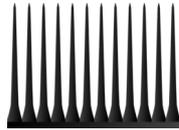
Peterson's algorithm oversimplified



Peterson's algorithm oversimplified

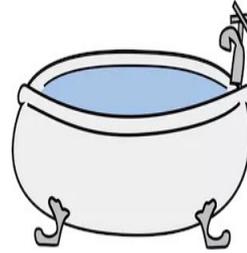
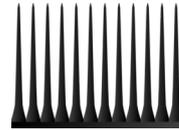
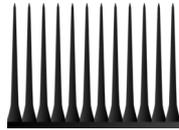


Peterson's algorithm oversimplified

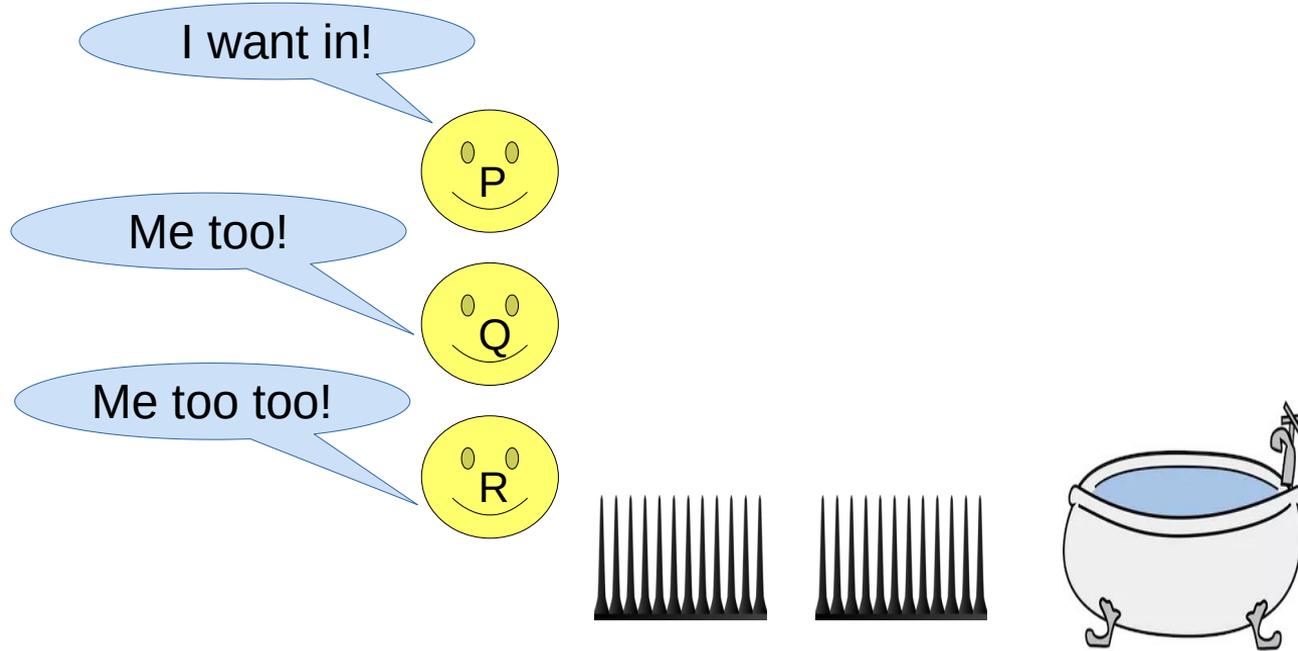


Make an obby with $n-1$ traps
for n processes.

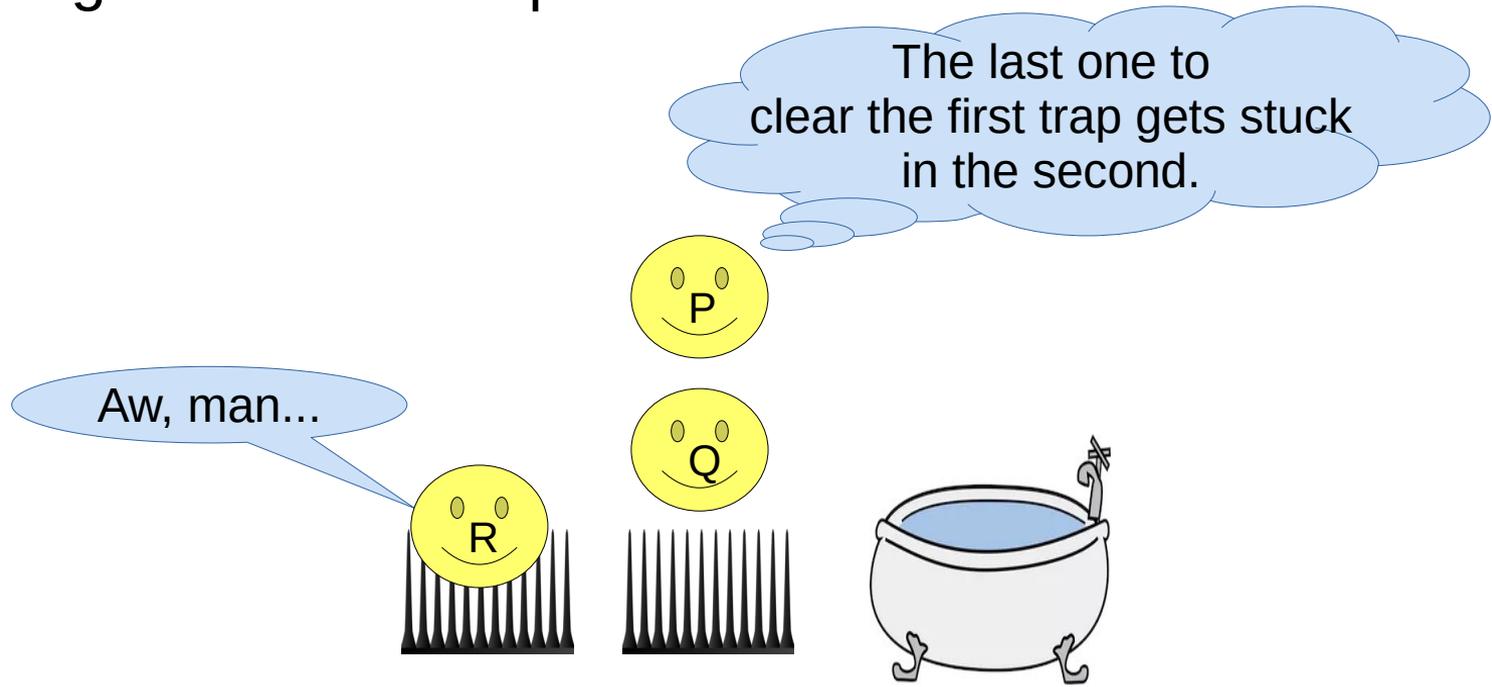
Peterson's algorithm oversimplified



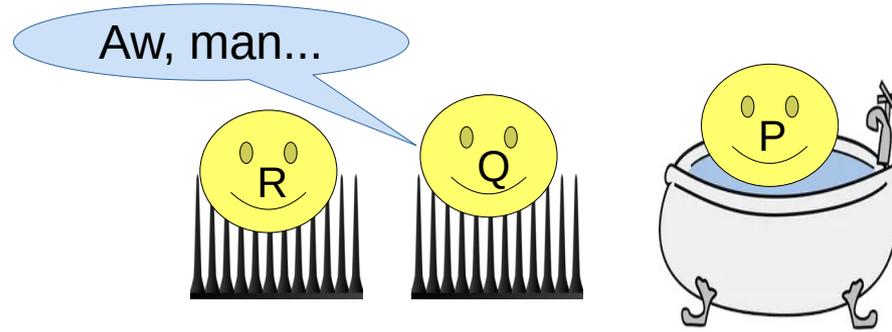
Peterson's algorithm oversimplified



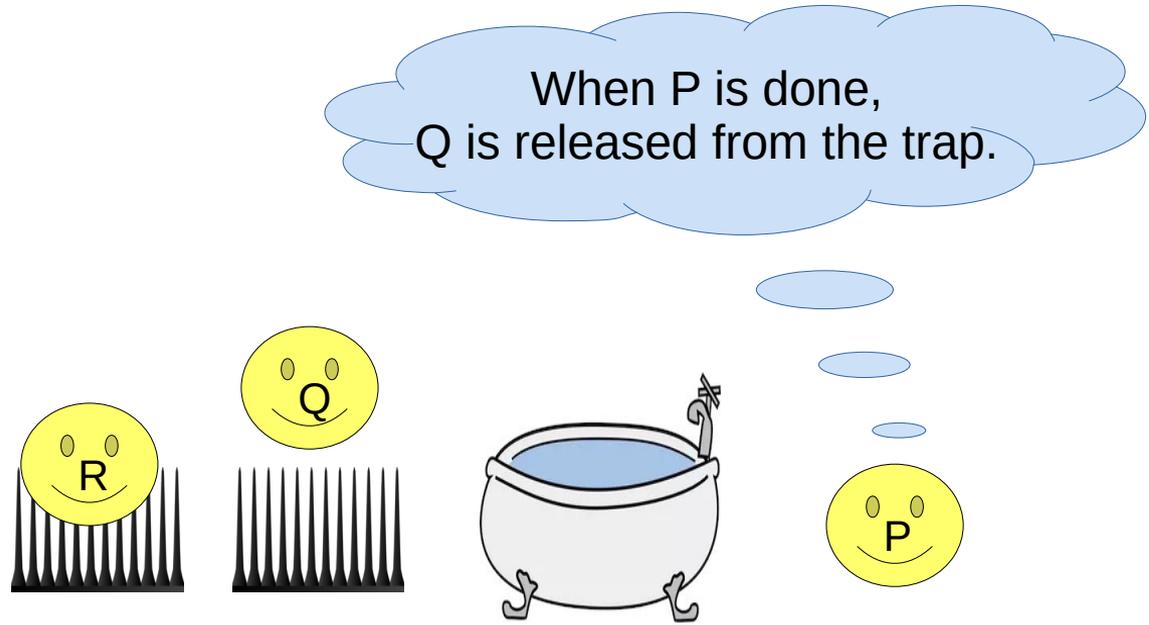
Peterson's algorithm oversimplified



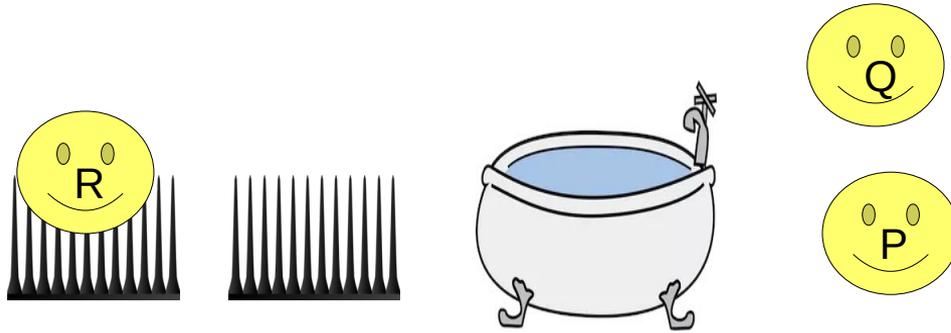
Peterson's algorithm oversimplified



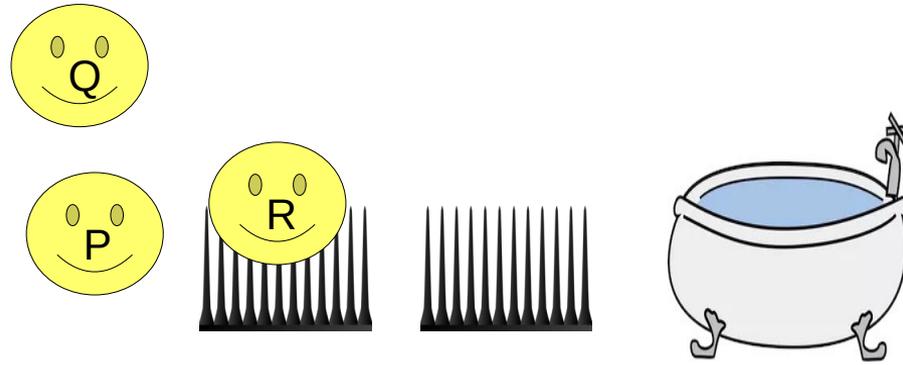
Peterson's algorithm oversimplified



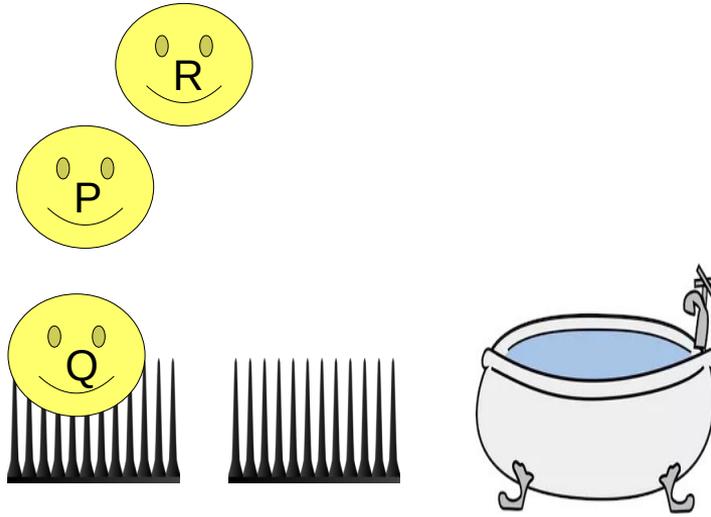
Peterson's algorithm oversimplified



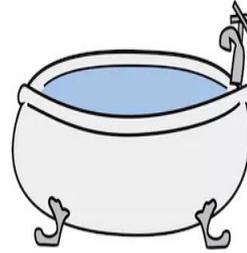
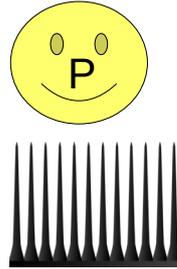
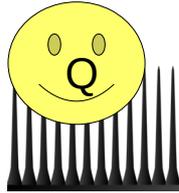
Peterson's algorithm oversimplified



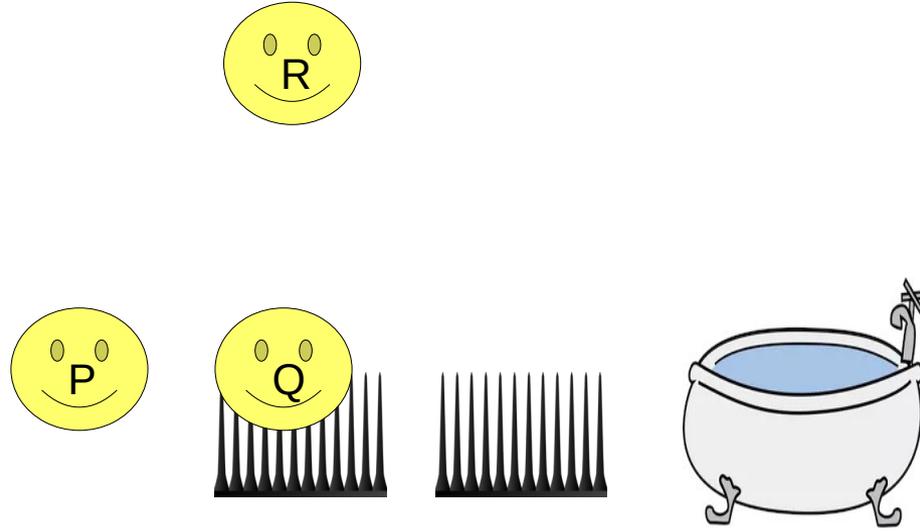
Peterson's algorithm oversimplified



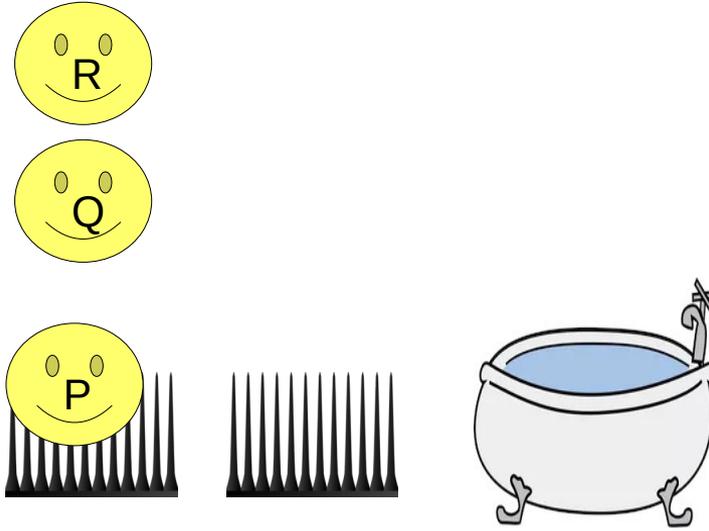
Peterson's algorithm oversimplified



Peterson's algorithm oversimplified



Peterson's algorithm oversimplified



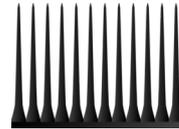
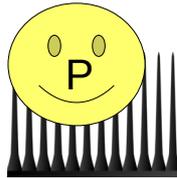
Peterson's algorithm oversimplified



By weak fairness,
R will eventually move to the next trap.

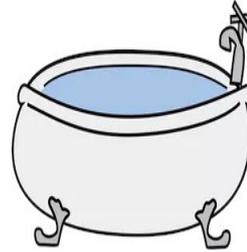
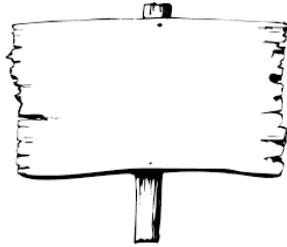
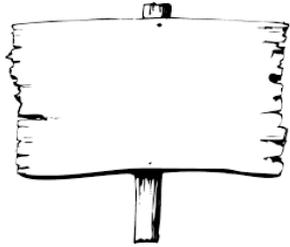
But we can't give a bound on
How many baths by P,Q that'll take:

Hence: eventual entry,
But not bounded wait



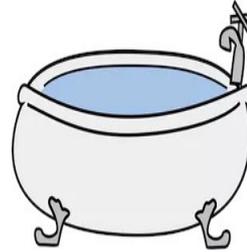
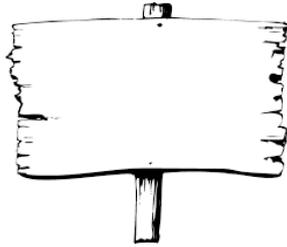
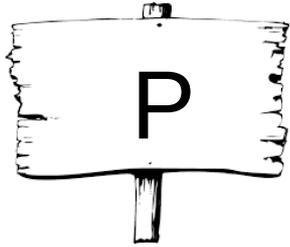
Lamport's fast algorithm oversimplified

When you pass a sign,
you write your name on it.

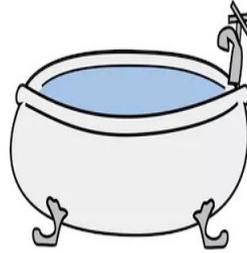
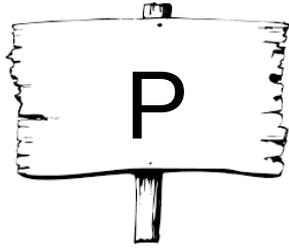
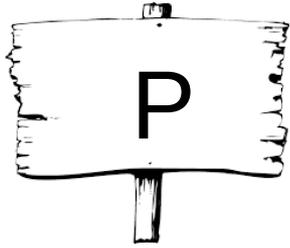


Lamport's fast algorithm oversimplified

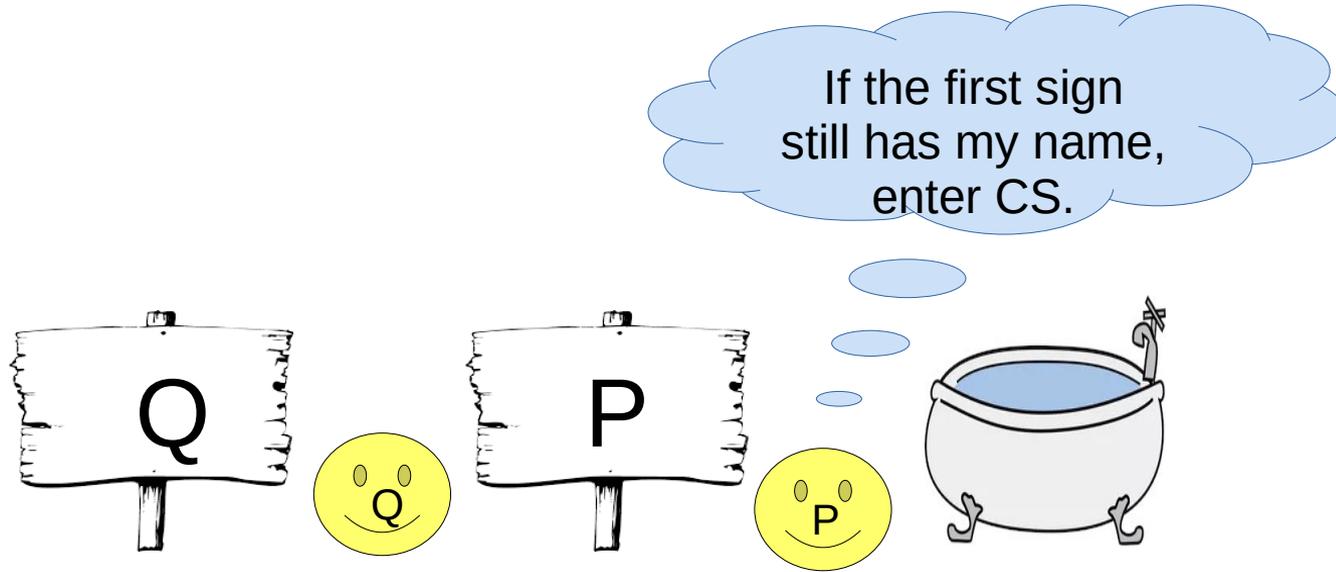
Look at the second sign.
Go back to start if there's
a name on it.



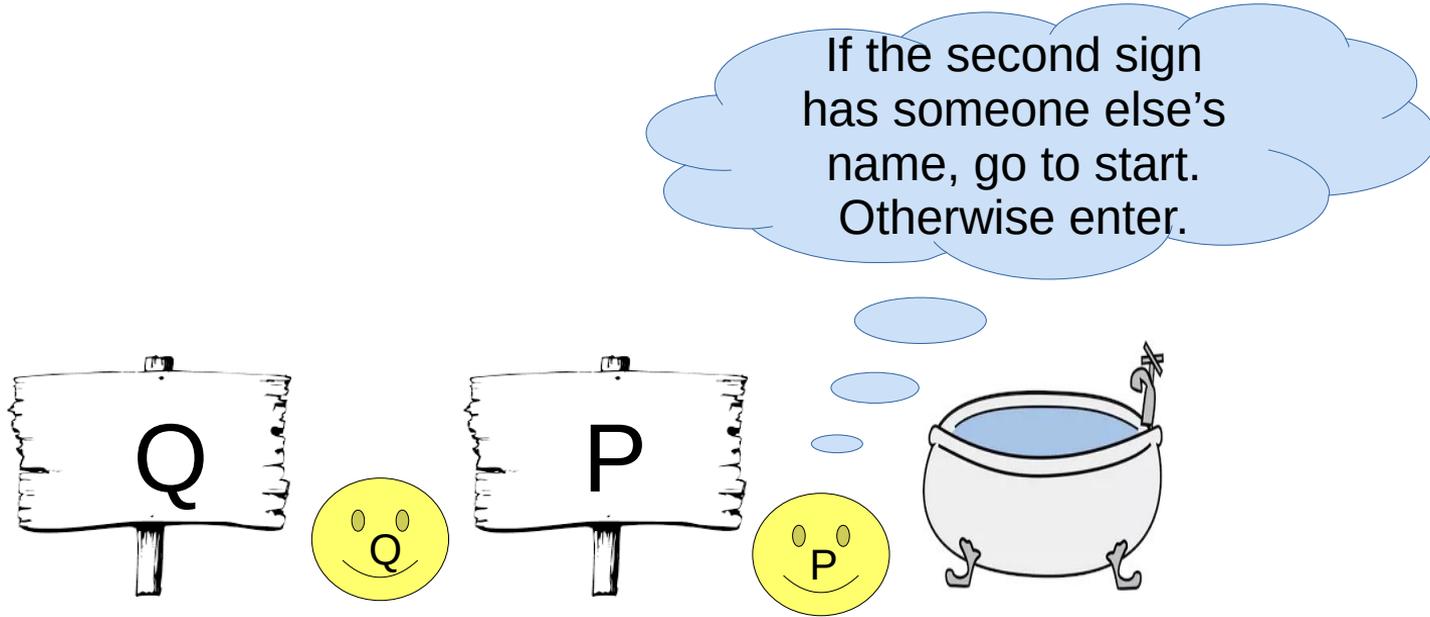
Lamport's fast algorithm oversimplified



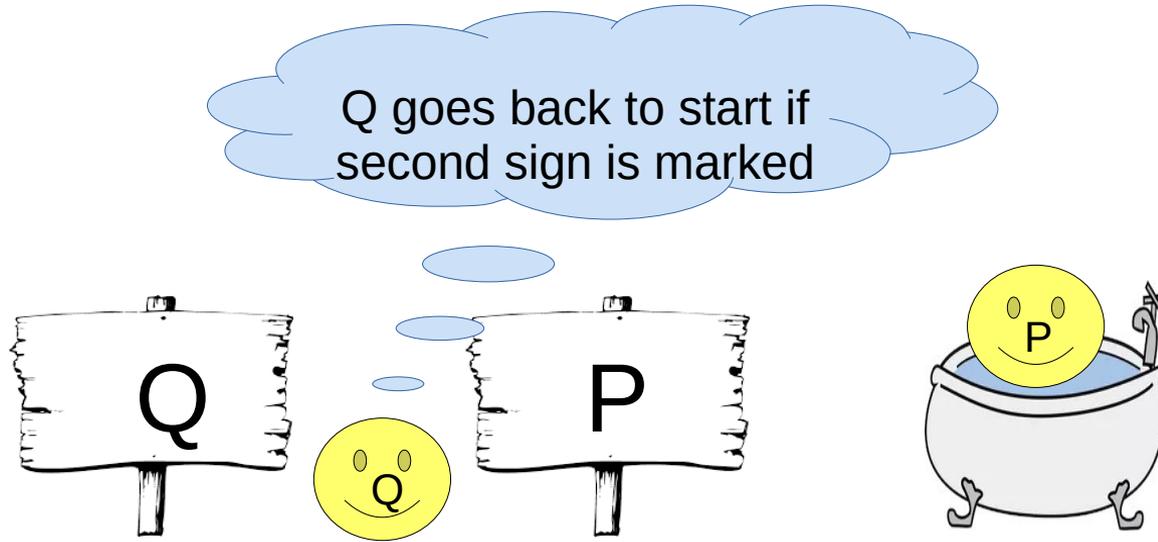
Lamport's fast algorithm oversimplified



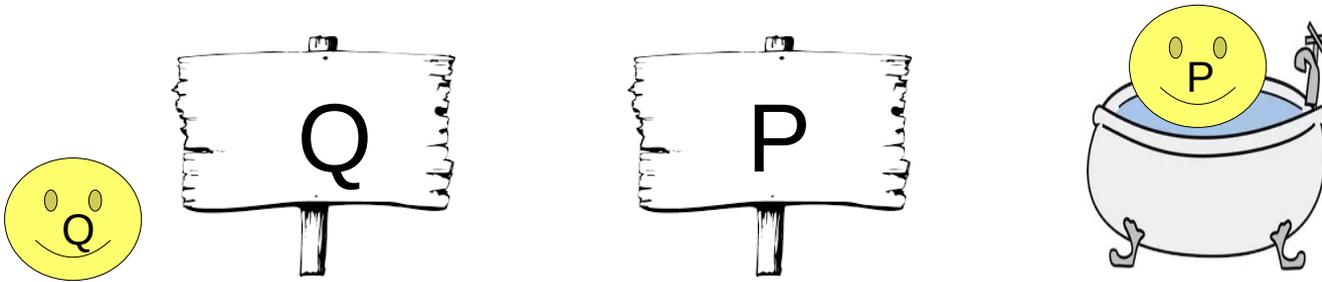
Lamport's fast algorithm oversimplified



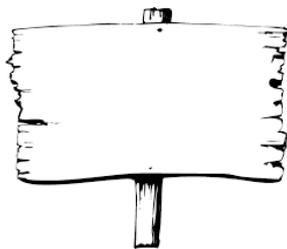
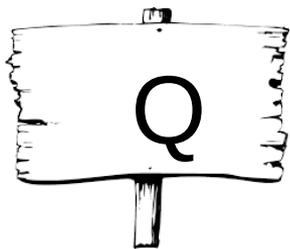
Lamport's fast algorithm oversimplified



Lamport's fast algorithm oversimplified

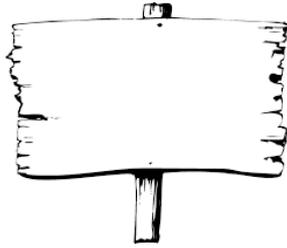
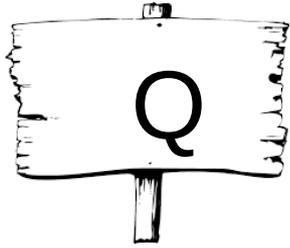


Lamport's fast algorithm oversimplified

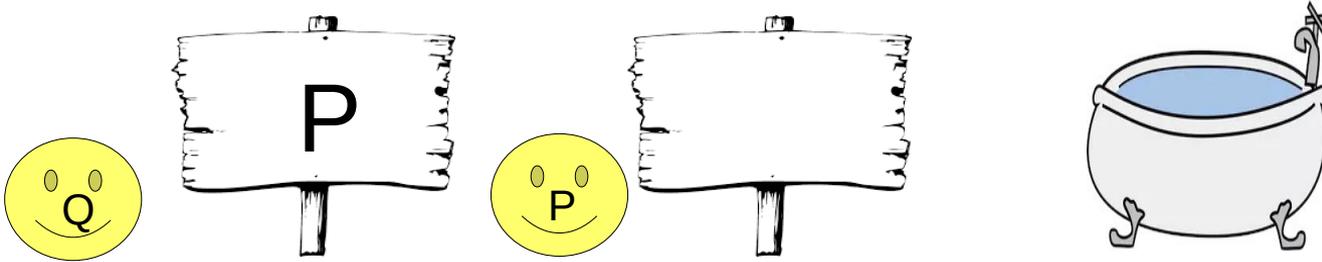


On the way out, remember to wipe the second sign!

Lamport's fast algorithm oversimplified

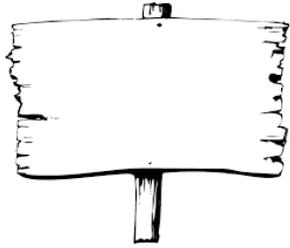


Lamport's fast algorithm oversimplified

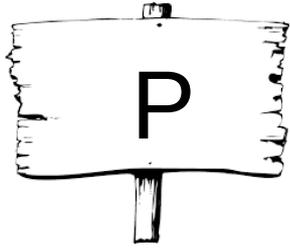


What if we have one sign?

Both procs
Read the sign
Before they have
Time to write it

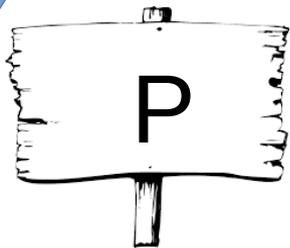


What if we have one sign?

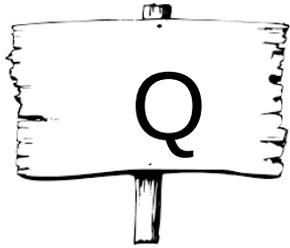


What if we have one sign?

Q read the sign
And saw -,
So can pass



What if we have one sign?



What if we have one sign?

