Salih SEVGİCAN 2013400219

CmpE 436 HW-4

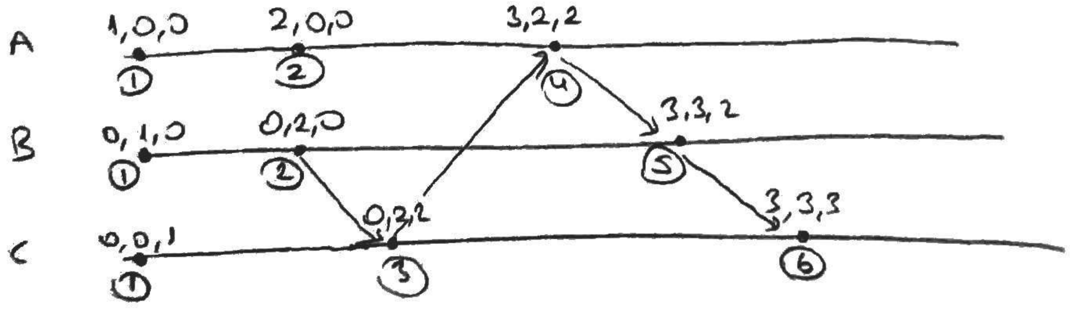
Question ①

There are a few approaches for solving this question.

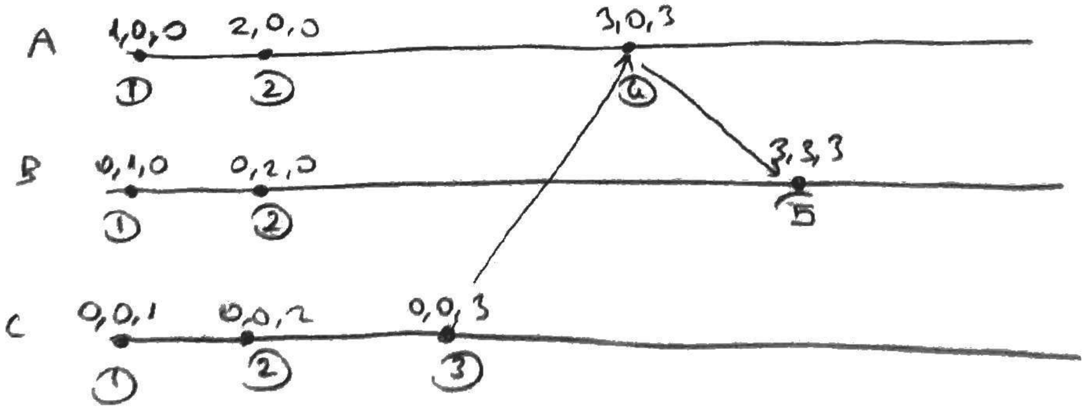
Logical clock is used to generate a total order trace among processes. Lamport's logical clock is one of these clocks. However, unlike the Lamport's logical clock, which ticks one by one for events, there are some other ways to implement logical clock and trace total order relation.

First of all lets consider following two examples.

Ex. 1



Ex. 2



As seen in the examples above, even if the vector clocks are same, there are 2 different logical clock for this case.

What if we assume that it is necessary to tick whenever a message passing or receiving event occurs? In this case we're able to convert a logical clock.

⟹

HW-4 Question 1 continues...

```
function convert (vector v)
        Logical_Clock <- max(v);
        for each element x in v do
            if (v[pid] == x)
                increment Logical_Clock
            end if
        endfor
        return Logical_Clock
end function.
```

The pseudo code above give logical clock.

The last approach is that it is enough for logical clock to generate a total order trace. For every vector clock among processes we can generate a custom logical clock (i.e. by summing up elements of vector) which can also grant us to trace total order.

For this approach we can simply write the following

```
function convert (vector v)
        return sum_of_elements(v)
end function.
```

Since the clock is constantly incremented by processes, this logical clock helps us to trace total order.

Salih Sevgican
2013400219