Mattia Danese
Homework #2
CS118 - Cloud Computing
Professor Sambasivan

Question 1

A. The probability of that a search query will take more than 1ms is the equal to the 1 minus the probability that a search query will not take more than 1ms

$P_{>1ms} = 1 - P_{\leq 1ms}$

Since there are 1520 total processes and each process has a probability of 0.999 of having a latency of 1ms, then the probability of all 1520 having a latency of 1ms is the product of every individual process' probability of having a latency of 1ms.

$P_{\leq 1ms} = (.999)^{1520}$

We can now substitute the above value for $P_{\leq 1ms}$ and solve for $P_{>1ms}$

$P_{>1ms} = 1 - (.999)^{1520}$

$P_{>1ms} \approx 0.781$

Thus, the probability that a search query will take more than 1ms is about 0.781.

B. One technique to reduce tail latency is process duplication. In process duplication, a process is duplicated and all duplications of that process are run concurrently, racing to finish first. The first duplication to finish is treated like the "main" process and is returned while the other duplications are terminated. Process duplication hedges on the assumption that the more processes running, the more likely it is for a process to finish with no added latency. Using Google's 99.9% latency of 1ms, if a process was duplicated just once, so now two of the same processes are concurrently running (independently), then the probability of *both* processes taking more than 1ms would be:

$P_{>1ms} = P'_{>1ms}$ and $P''_{>1ms} = P'_{>1ms} * P''_{>1ms} = (0.001)^2 = 0.000001$

Thus, process duplication greatly decreases the probability of a process taking more than 1ms to complete.

Question 2

A. The definition of Lamport's *Happens-Before* relationship is:

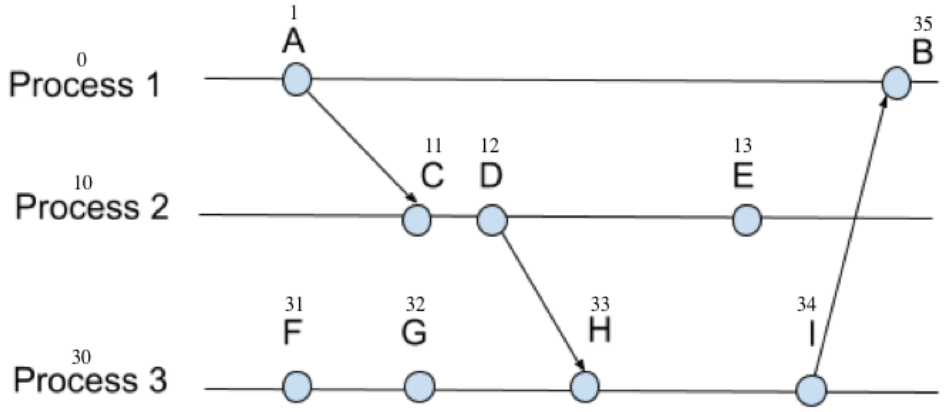
If some event *a* on process *x* has a Lamport timestamp that is less than some other event *b* on process *y*, namely $L(a) < L(b)$, then:

1. process *x* may have completed event *a* and then communicated with process *y* before process *y* began event *b*, in which event *a* '*happens-before*' event *b*, or
2. event *a* and event *b* happened concurrently

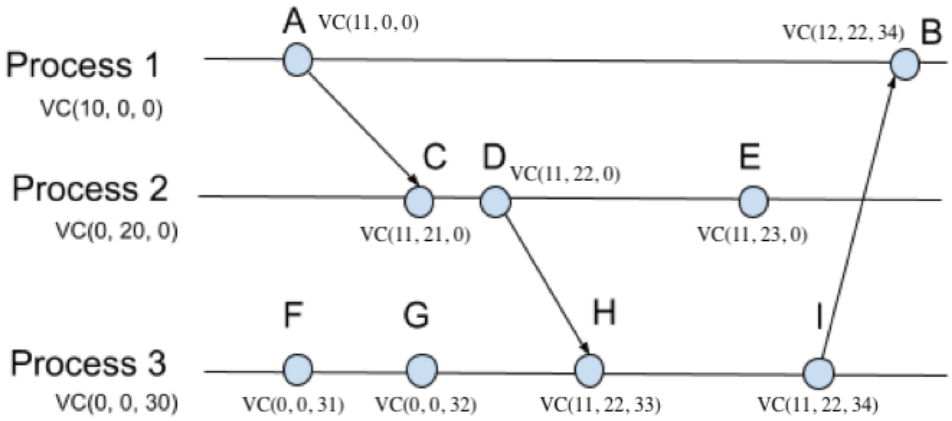
B. The following groups of events can be deemed concurrent:

- A and F
- A and G
- C and F
- C and G
- D and F
- D and G
- E and B
- E and F
- E and G
- E and H
- E and I

C.



D.


A VC(11, 0, 0)
VC(12, 22, 34) B
Process 1
VC(10, 0, 0)
C
D VC(11, 22, 0)
E
Process 2
VC(0, 20, 0)
VC(11, 21, 0)
VC(11, 23, 0)
F
G
H
I
Process 3
VC(0, 0, 30)
VC(0, 0, 31)
VC(0, 0, 32)
VC(11, 22, 33)
VC(11, 22, 34)

Question 3

Even though modern computers can associate GPS timestamps with individual events, solving the synchronized clock problem that logical clocks also solve, the happens-before relationship is still useful because it can imply potential causality between events. GPS timestamps are just a pinpoint in time, so one can definitively assert if one process occurs before another, though there is no additional information regarding if the processes are in any way linked or if they are completely independent of each other and one just happened to occur before another. Happens-before relationships, on the other hand, may be less definitive in asserting if one process occurs before another, but they do allude to which processes in the past may have had an impact on current or future processes.