Assignment 1.1 Quick Quiz.

Answer the following questions:

1. Can a happened-before diagram depict several executions of a distributed system or only one?

2. Can a single happened-before diagram illustrate all the executions (runs) that a synchronization algorithm can perform?

Assignment 1.2 Happened-Before Diagram

For each of the following diagrams, decide if they are valid happened-before diagrams. Prove your answer by defining a mapping $C : E \rightarrow \mathbb{N}$ that satisfies the clock condition or by showing that no such mapping exists. Here $E = \{e_1, \ldots , e_{15}\}$ is the set of events. (The clock condition states that for all $p_i, p_j \in P$, if $p_i$ happens before $p_j$ then $C(p_i) < C(p_j)$.)

- Diagram one:

```
  P e1  e2  e3  e4  e6
  Q  e5
  R e7  e8  e9
  S e10 e11
  T e12 e13 e14 e15
```

- Diagram two: as the diagram of 1., but with the arrow between $e_6$ and $e_7$ pointing in the opposite direction
Assignment 1.3 Concurrent events

1. List for Diagram one from above three pairs of concurrent events. Use the notion \((e_i, e_j)\) for \(e_i\) and \(e_j\) are concurrent events.

2. Proof by example that the following statements do not hold:
   - If \(e_1\) and \(e_2\) are concurrent and \(e_2\) and \(e_3\) are concurrent then \(e_1\) and \(e_3\) are concurrent. (With other words: If \((e_1, e_2)\) and \((e_2, e_3)\) then \((e_1, e_3)\).)
   - If \(e_1\) and \(e_2\) are concurrent and \(e_2\) happened before \(e_3\) then \(e_1\) and \(e_3\) are concurrent. (With other words: If \((e_1, e_2)\) and \(e_2 \rightarrow e_3\) then \((e_1, e_3)\).)
Consider a distributed system with CPUs A, B, C. For a cache line $z$ in the cache of CPU A explain the transition $b(E \rightarrow M), f(M \rightarrow S), h(I \rightarrow M), i(E \rightarrow I)$, from one state $s \in \{M, E, S, I\}$ to another state $s' \in \{M, E, S, I\}$. Which messages (Read (Response), Invalidate (Acknowledge), Read Invalidate, Writeback (Read Response)) are sent between the CPUs?