**Q-1: Describe the dining philosopher’s problem?**

The Dining Philosopher Problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each philosopher. A philosopher may eat if he can pickup the two chopsticks adjacent to him. One chopstick may be picked up by any one of its adjacent followers but not both.

**Q-4: What are other ways which you have studied to solve this problem?**

**Resource hierarchy solution**

This solution to the problem is the one originally proposed by [Dijkstra](https://en.m.wikipedia.org/wiki/Edsger_W._Dijkstra). It assigns a [partial order](https://en.m.wikipedia.org/wiki/Partially_ordered_set) to the resources (the forks, in this case), and establishes the convention that all resources will be requested in order, and that no two resources unrelated by order will ever be used by a single unit of work at the same time.

**Arbitrator solution**

Another approach is to guarantee that a philosopher can only pick up both forks or none by introducing an arbitrator, e.g., a waiter. In order to pick up the forks, a philosopher must ask permission of the waiter. The waiter gives permission to only one philosopher at a time until the philosopher has picked up both of their forks.

**Chandy/Misra solution**

In 1984, [K. Mani Chandy](https://en.m.wikipedia.org/wiki/K._Mani_Chandy) and [J. Misra](https://en.m.wikipedia.org/wiki/Jayadev_Misra) proposed a different solution to the dining philosophers problem to allow for arbitrary agents (numbered *P*1, ..., *Pn*) to contend for an arbitrary number of resources, unlike Dijkstra's solution. It is also completely distributed and requires no central authority after initialization.