HOMEWORK REPORT

In this homework, we tried to find a solution about "Dining Philosophers" problem. We used "Chandy/Misra" solution. In this report, we are going to explain our solution method.

```
def eat_chandy_misra(self):
with self.left_fork.lock:
self.left_fork.request(self.index)
with self.right_fork.lock:
    self.right_fork.request(self.index)
if self.left_fork.owner == self.index and self.right_fork.owner == self.index:
    self.left_fork.grant()
    self.right_fork.grant()
    self.eat()
else:
    self.left_fork.release()
```

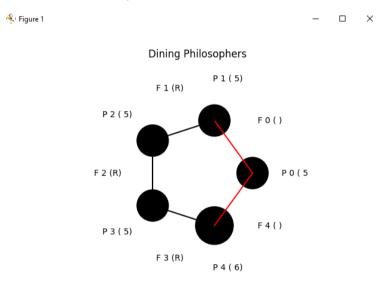
This is our Chandy/Misra function. There are some steps in this function above.

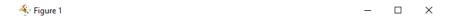
- 1. Acquiring the Left Fork: The philosopher first attempts to acquire the left fork's lock using a with statement (with self.left_fork.lock:). This ensures that access to the left fork is properly synchronized.
- 2. Requesting the Left Fork: The philosopher then requests the left fork by calling the request(self.index) method on the left fork object. The philosopher's index is passed as an argument to identify itself.
- 3. Acquiring the Right Fork: Next, the philosopher attempts to acquire the right fork's lock using a nested with statement (with self.right_fork.lock:).
- 4. Requesting the Right Fork: Similar to the left fork, the philosopher requests the right fork by calling self.right fork.request(self.index).

- 5. Checking Ownership of Forks: The philosopher checks if it successfully owns both the left and right forks (if self.left_fork.owner == self.index and self.right_fork.owner == self.index:).
- 6. Granting Forks and Eating: If the philosopher owns both forks, it grants ownership of both forks (self.left_fork.grant() and self.right_fork.grant()) and proceeds to eat by calling the self.eat() method.
- 7. Releasing Forks if Ownership Failed: If the philosopher couldn't own both forks, it releases any acquired forks (self.left_fork.release() and self.right_fork.release()) and doesn't proceed to eat.

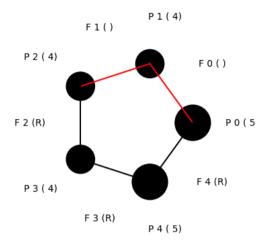
x=-0.249 y=0.051

Here are some outputs on this solution



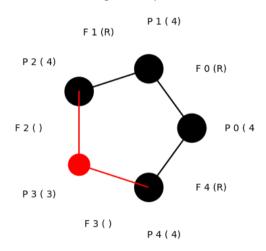


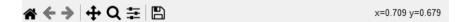
Dining Philosophers

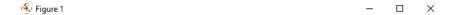




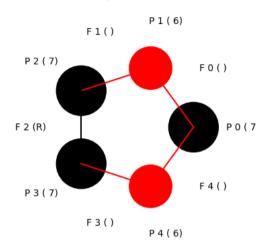
Dining Philosophers

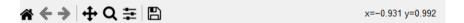






Dining Philosophers





Kadir KARADUMAN 190315005

Evren YALNIZ 190315019

Berke YILDIZ 190315048