

### Умова:

Розробити та запрограмувати алгоритм вирішення задачі про філософів, що обідають – [https://en.wikipedia.org/wiki/Dining\\_philosophers\\_problem](https://en.wikipedia.org/wiki/Dining_philosophers_problem) .  
Рішення має враховувати:

- **ексклюзивність** ресурсів (виделку як ресурс може взяти лише один (з двох adjacent neighboring) філософ у кожний момент часу, і тільки якщо вона вільна)
- deadlock-free (щоб система не могла стати в **тупик**)
- livelock-free (щоб філософи не померли від голодування, коли їм довго не вдається взяти виделки, бо вони якимось чином блокуються сусідами), ситуація **нескінченного очікування**

Перевага надається рішенням, які:

- реалізують однаковий алгоритм дій усіх філософів (не залежать від номера філософа, тощо)
- не мають централізованої сутності (офіціанта, оракула, тощо), який керує розподілом ресурсів-виделок
- не передбачають комунікацію між філософами (щоб вони домовились про виделки чи інші спільні дії)
- є масштабованим на будь-яку кількість філософів (і виделок відповідно) і не мають очевидних bottlenecks
- можна довести формально їх коректність і виконуваність всіх вказаних вимог

Всі програми-філософи мають діяти за однаковим алгоритмом. Всі вони рівноправні (і мають бути взаємозамінні), жоден не має пріоритету над іншими.

### Рішення:

До основних проблем належать

- **Deadlock** – коли всі філософи взяли по одній вилці й відмовляються йти на компроміс.

*В нашому рішенні проблема уникатиметься тим, що за відсутності доступу до іншої вилки філософи кластимуть її назад.*

- **Livelock, Starvation** – коли філософи беруть та кладуть вилки але ніхто не їсть і в результаті частина помирає від голоду.

*Метод полягає в мінімізації шансів голодної смерті за рахунок того, що чим голодніший філософ, тим впертіше він чекає на другу вилку перш ніж покласти ту що має назад на деякий час.*

Маємо наступний алгоритм:

```
int t = timer.new()
int r = rand()
while true
{
    sleep(r)
    wait(left_fork)
    if wait_with_timeout(right_fork, t+r)
    {
        eat()
        release_forks()
        think()
        t = timer.new()
    }
    else
    {
        signal(left_fork)
        r = rand()
    }
}
```

t – змінна для слідкування за мірою голодності філософа, після того як він бере вилку він чекає t часу на іншу, тобто голодніші філософи мають більше шансів поїсти.

r – змінна для внесення розсинхронізації, наприклад, щоб уникнути того що два сусіди працюють в унісон і потребують вилок одночасно.

Після того як філософ поїв t обнуляється, якщо поїсти не вдалося, філософ починає спочатку: почекати r часу, взяти ліву вилку, чекати на праву і т. д.

## Тестування:

Маємо наступні параметри:

```
int numberOfDishes = 20;
const int EATING_TIME = 500; //milliseconds
const int STARVATION_TIME = 2000;
const int NUMBER_OF_PHILOSOPHERS = 999;
```

Та мій метод у порівнянні з очевидно недосконалими:

Livelock solution:

Філософ чекає 5 мілісекунд перш ніж покинути вилку

```
philosopher 662 is eating...
philosopher 132 is eating...
philosopher 662 is thinking about those 0 plates of pasta left to eat...
philosopher 298 is thinking about those 0 plates of pasta left to eat...
philosopher 421 is thinking about those 0 plates of pasta left to eat...
philosopher 78 is thinking about those 0 plates of pasta left to eat...
philosopher 42 is thinking about those 0 plates of pasta left to eat...
philosopher 244 is thinking about those 0 plates of pasta left to eat...
philosopher 194 is thinking about those 0 plates of pasta left to eat...
philosopher 508 is thinking about those 0 plates of pasta left to eat...
philosopher 447 is thinking about those 0 plates of pasta left to eat...
philosopher 468 is thinking about those 0 plates of pasta left to eat...
philosopher 623 is thinking about those 0 plates of pasta left to eat...
philosopher 775 is thinking about those 0 plates of pasta left to eat...
philosopher 917 is thinking about those 0 plates of pasta left to eat...
philosopher 90 is thinking about those 0 plates of pasta left to eat...
philosopher 408 is thinking about those 0 plates of pasta left to eat...
philosopher 647 is thinking about those 0 plates of pasta left to eat...
philosopher 585 is thinking about those 0 plates of pasta left to eat...
philosopher 259 is thinking about those 0 plates of pasta left to eat...
philosopher 132 is thinking about those 0 plates of pasta left to eat...
philosopher 904 is thinking about those 0 plates of pasta left to eat...
philosopher 104 is thinking about those 0 plates of pasta left to eat...
00:00:15.5853467
starved philosophers: 19
```

50 мілісекунд:

```
philosopher 717 is eating...
philosopher 905 is thinking about those 0 plates of pasta left to eat...
philosopher 936 is eating...
philosopher 143 is thinking about those 0 plates of pasta left to eat...
philosopher 437 is thinking about those 1 plates of pasta left to eat...
philosopher 584 is thinking about those 0 plates of pasta left to eat...
philosopher 66 is thinking about those 0 plates of pasta left to eat...
philosopher 883 is thinking about those 0 plates of pasta left to eat...
philosopher 936 is thinking about those 0 plates of pasta left to eat...
philosopher 437 is eating...
philosopher 891 is thinking about those 0 plates of pasta left to eat...
philosopher 752 is thinking about those 0 plates of pasta left to eat...
philosopher 437 is thinking about those 0 plates of pasta left to eat...
philosopher 717 is thinking about those 1 plates of pasta left to eat...
philosopher 717 is eating...
philosopher 717 is thinking about those 0 plates of pasta left to eat...
00:00:16.5471061
starved philosophers: 26
```

Deadlock solution:

Філософи відмовляються повертати вилку:

```
philosopher 259 is thinking about those 1 plates of pasta left to eat...
philosopher 257 is eating...
philosopher 258 is thinking about those 0 plates of pasta left to eat...
philosopher 260 is thinking about those 0 plates of pasta left to eat...
philosopher 259 is eating...
philosopher 257 is thinking about those 0 plates of pasta left to eat...
philosopher 256 is eating...
philosopher 259 is thinking about those 0 plates of pasta left to eat...
philosopher 256 is thinking about those 0 plates of pasta left to eat...
philosopher 255 is eating...
philosopher 254 is eating...
philosopher 255 is thinking about those 0 plates of pasta left to eat...
philosopher 254 is thinking about those 0 plates of pasta left to eat...
00:00:59.4501776
starved philosophers: 0
```

Ніхто не помер і всі поїли, deadlock не стався, але цей процес зайняв у 4 рази більше часу ніж попередній алгоритм.

Рішення зі змінною t:

```
philosopher 580 is thinking about those 0 plates of pasta left to eat...
philosopher 329 is thinking about those 0 plates of pasta left to eat...
philosopher 612 is thinking about those 1 plates of pasta left to eat...
philosopher 612 is eating...
philosopher 612 is thinking about those 0 plates of pasta left to eat...
00:00:18.0888701
starved philosophers: 0
```

Філософи їдять швидко і ніхто не помирає з голоду.

Я пробував додати та відняти кількість філософів та тарілок, неочікуваних результатів не отримав.