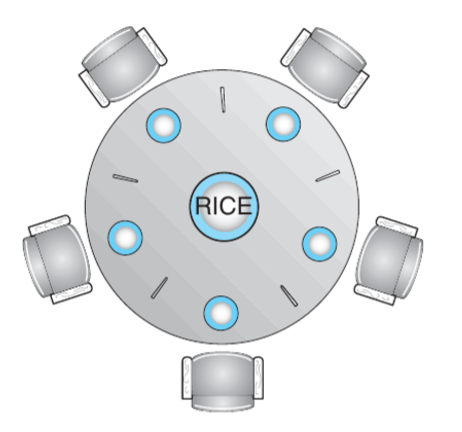
[**Dining philosophers**](https://gist.github.com/Alexey-N-Chernyshov/16198f75b284191bf2406d27eaad6b23)

Có 5 triết gia, 5 chiếc đũa, 5 bát cơm và một âu cơm được bố trí như hình vẽ:



Các triết gia chỉ làm 2 việc: Ăn và suy nghĩ

Suy nghĩ: Không ảnh hưởng đến các triết gia khác, đũa, bát và âu cơm.

Ăn: Mỗi triết gia phải có đủ 2 chiếc đũa gần nhất ở bên phải và bên trái mình; chỉ được lấy 1 chiếc đũa một lần và không được phép lấy đũa từ tay triết gia khác.

Khi ăn xong: Triết gia bỏ cả hai chiếc đũa xuống bàn và tiếp tục suy nghĩ.

public void run() {  
 while (true) {  
 leftFork.grab();  
 rightFork.grab();  
 eat();  
 leftFork.release();  
 rightFork.release();  
 }  
}

Mã lệnh này có thể gây bế tắc (deadlock) nếu cả 5 triết gia đều lấy được 1 chiếc đũa và chờ để lấy chiếc còn lại nhưng không bao giờ lấy được!

Giải pháp. Cho các triết gia là để lấy lock theo 1 thứ tự cố định.

* Cho phép các triết gia lấy theo 1 thứ tự trái -> phải

if(left.getId()<right.getId()){  
 first=left;second=right;  
}else{  
 first=right;second=left;  
}

* Chỉ có nhiều nhất 4 triết gia đồng thời lấy đũa , dẫn đến có ít nhất 1 triết gia lấy đc 2 chiếc đũa.

if (i == philosophers.length - 1) {  
// The last philosopher picks up the right fork first  
 philosophers[i] = new Philosopher(rightFork, leftFork);  
} else {  
 philosophers[i] = new Philosopher(leftFork, rightFork);  
}

* Sử dụng giải pháp bất đối xứng: Triết gia mang số lẻ lấy chiếc đũa đầu tiên ở bên trái, sau đó chiếc đũa ở bên phải; triết gia mang số chẵn lấy chiếc đũa đầu tiên ở bên phải, sau đó lấy chiếc đũa bên trái.