class TSPCONST():

def tour\_cost(self,instance, solution, problem\_size): -> cost

# Nhận vào 1 bảng tọa độ (instance), 1 solution với proble, tính cost của nó

def generate\_neighborhood\_matrix(self,instance):

# Nhận vào 1 instance, cho ra bảng n hàng, với hàng i là các đỉnh gần với đỉnh I theo thứ tự gần đến xa

def greedy(self,eva):

# nhận vào 1 heurictic eva, tính cost trung bình của nó trên 1 tập các instance để đánh giá độ tốt

def evaluate(self, code\_string):

# nhận vào 1 đoạn code dưới dạng string, chạy hàm greedy với code đó, trả về đánh giá( fitness).

Selection:  
 **Chọn ngẫu nhiên (Random Selection)**

* Mỗi cá thể trong quần thể có xác suất bằng nhau được chọn.
* Không dựa vào độ tốt (fitness).

 **Chọn theo hạng (Rank-based Selection) (mặc định dùng)**

* Sắp xếp cá thể theo giá trị fitness (hoặc objective).
* Cá thể có hạng cao hơn được gán xác suất chọn cao hơn.

 **Chọn theo tỉ lệ fitness (Fitness-Proportionate / Roulette Wheel Selection)**

* Xác suất chọn tỉ lệ thuận với giá trị fitness (hoặc nghịch với objective).
* Cá thể tốt hơn có nhiều cơ hội được chọn hơn.

 **Giải đấu (Tournament Selection)**

* Lấy ngẫu nhiên một nhóm nhỏ (ví dụ 2 hoặc 3 cá thể).
* So sánh fitness trong nhóm, chọn cá thể tốt nhất làm cha.

Management:  
Hàm population\_management (pop-greedy) có nhiệm vụ:

1. Loại bỏ cá thể lỗi (objective=None).
2. Loại bỏ cá thể trùng lặp theo giá trị objective.
3. Giữ lại số lượng cá thể tốt nhất (có objective nhỏ nhất) bằng heapq.nsmallest.

Flow tổng quát :  
runEOH.py -> EVOL.run(): get\_problem(), get\_Methods() -> EOH.run()

Trong EOH.run()

+) Khởi tạo population ( pop init)

+) Lặp tiến hóa: Với mỗi vòng lặp, chạy qua hết các operator, mỗi operator có 1 xác suất kích hoạt

+) Với mỗi operator được kích hoạt gọi hàm interface\_ec.get\_algorithm(population, op) để lấy về offspring (con), rồi thêm vào population, sau đó lọc với manage.population\_management(population, size\_act)

Trong interface\_ec.get\_algorithm(population, op) -> get\_offspring ()

Trong get\_offspring () có

\_get\_alg(pop, operator)

Trong \_get\_alg(pop, operator): Gọi hàm select lấy parents, sau đó gọi evol.\*op(parents)

Trong evol.\*op(): Lấy prompt\_content, truyền qua interface\_llm.get\_response(prompt\_content) lấy response, parse code và miêu tả thuật toán ra.

Lấy ra fitness của offspring bằng hàm evaluate của class TSPCONST():