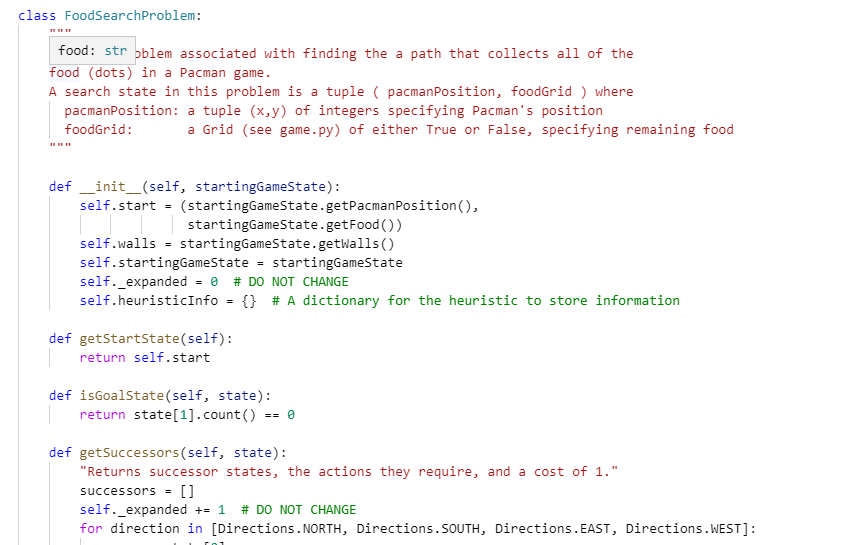
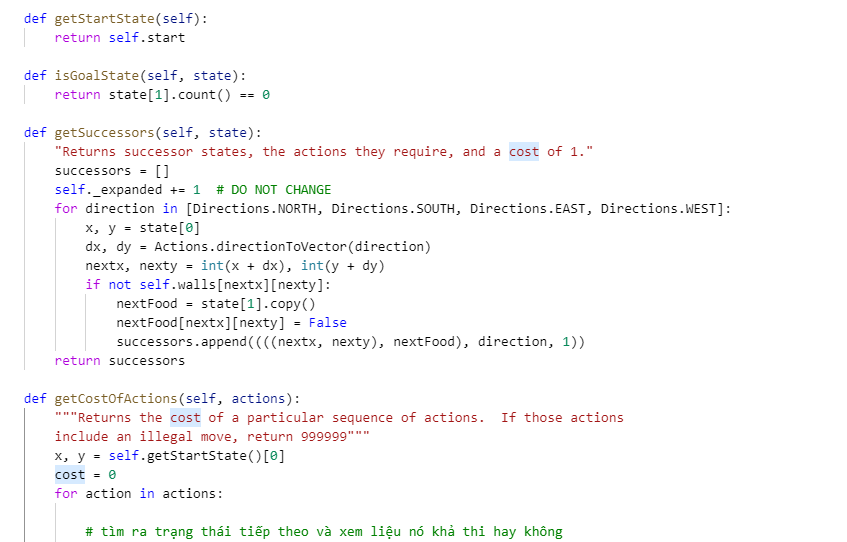
**BÁO CÁO KẾT QUẢ THUẬT TOÁN**

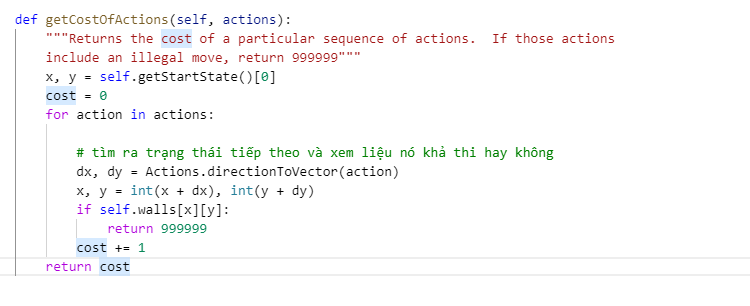
1. **Heuristic cho bài toán ăn hết thức ăn**

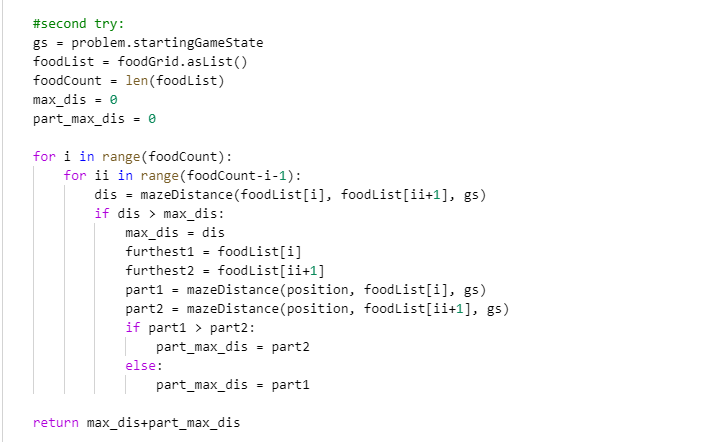
Trong phần bào toán này, mình sẽ thiết kế 1 class **FoodSearchProblem** trong file *searchAgent.py* để thể hiện bài toán tìm kiếm sau: tìm đường đi ngắn nhất cho Pacman trong mê cung sao cho Pacman đi qua tìm hết thức ăn.

-Hình ảnh minh hoạ cài Class mà cần làm:

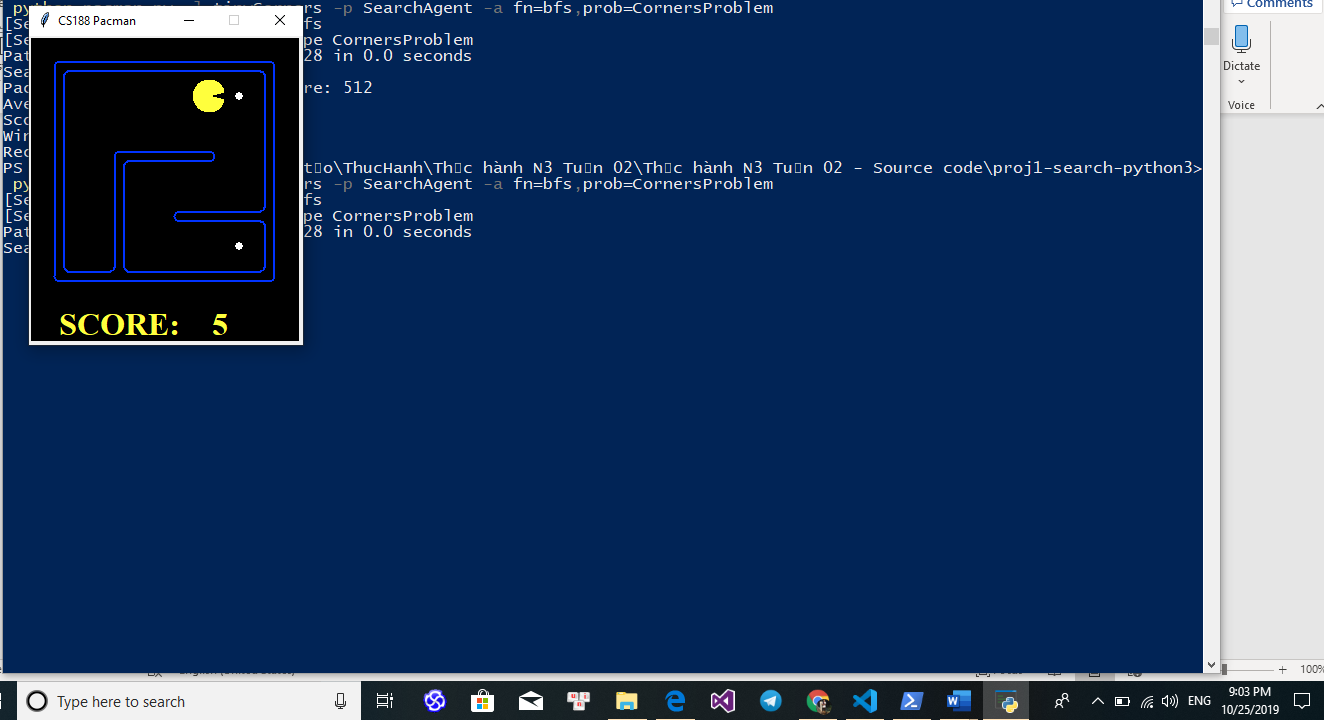


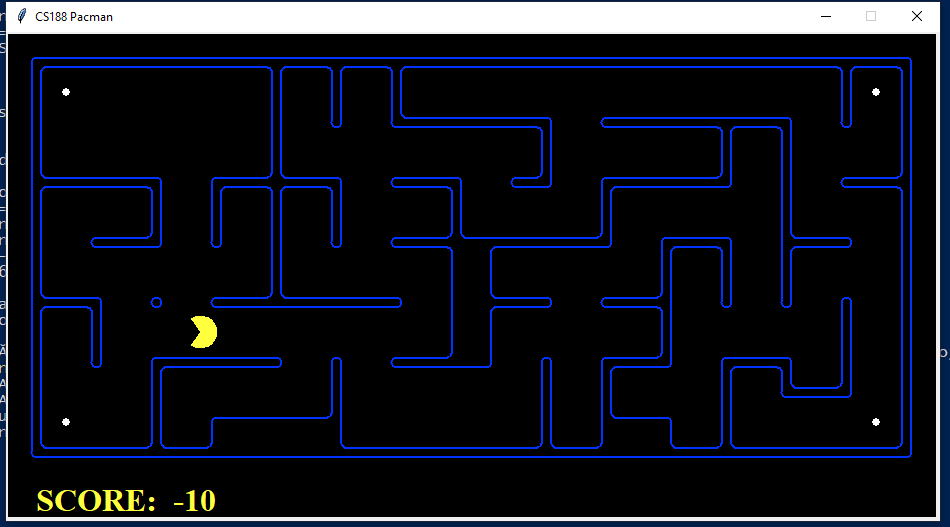




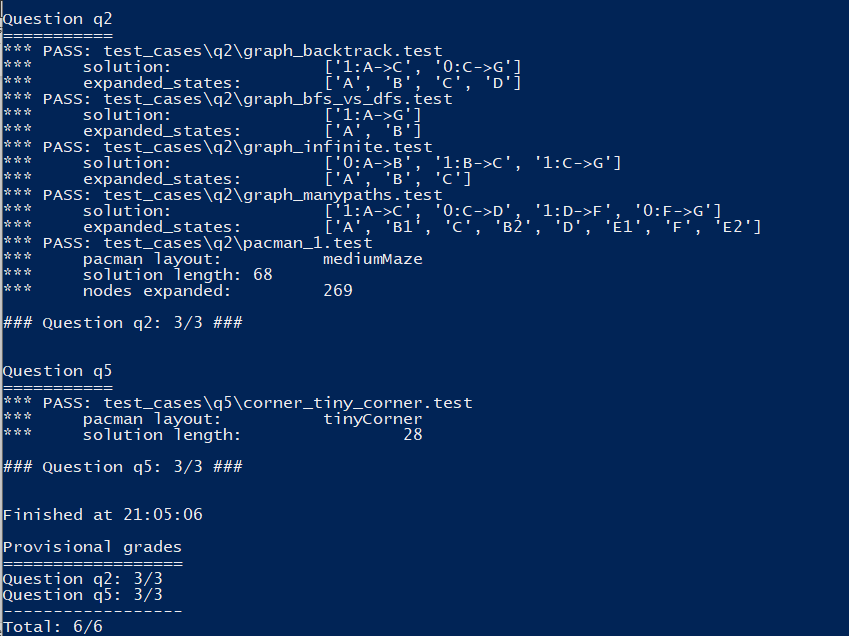


-Sau khi xây dựng code xong, mình sẽ chạy thực thi đoạn code qua lệnh:

***python pacman.py -l trickySearch -p AStarFoodSearchAgent với AStarFoodSearchAgent là “shortcut” của: -p SearchAgent -a "fn=astar,prob=FoodSearchProblem,heuristic=foodHeuristic"*** 



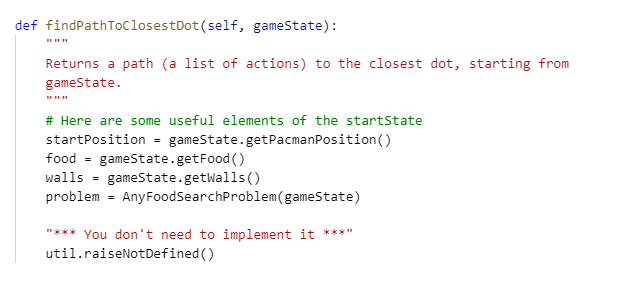
-Và bạn có thể bạn gõ ***python autograder.py -q q7*** để kiểm tra phần cài đặt của bạn với các bộ test khác nhau.



1. **Tìm kiếm nhanh**

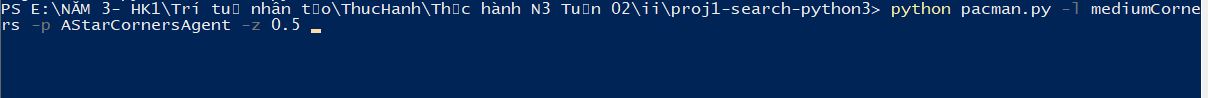
Sau khi hoàn thành câu 2.7 thì mình tiếp tục đến với thuật toán tìm kiếm nhanh bằng **Heuristic .**Mình sẽ viết hoàn chỉnh hàm **cornersHeuristic** trong file *searchAgent.py*. Hàm này nhận hai tham số đầu vào là trạng thái state và bài toán problem, và trả về chi phí ước lượng từ state đến đích.

-Hình ảnh minh hoạ code và note lại những ghi chú trong khi code:



-Sau khi đã cài đặt xong, bạn có thể kiểm tra một cách trực quan trên game Pacman bằng cách gõ câu lệnh:

***python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5***



-Sau khi chạy lệnh , bạn có thể gõ ***python autograder.py -q q8*** để kiểm tra phần cài đặt của bạn với các bộ test khác nhau:

--🡪 Kết quả này bị Fail rồi, mình sẽ cố gắng sửa lại thuật toán sớm nhất có thể !

THANK YOU