**THUẬT GIẢI DI TRUYỀN CHO BÀI TOÁN KNAPSACK**

Năm học: 2015-2016

Họ vs Tên: Phạm Thị Cẩm Duyên

MSSV: 1312078

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**Bài toán**:

Bài toán Knapsack: cho một chiếc túi có khả năng chứa khối lượng tối đa . Có vật có khối lượng lần lượt là và giá trị tương ứng là . Tìm tập con có giá trị nhất mà chiếc túi có khả năng mang được.

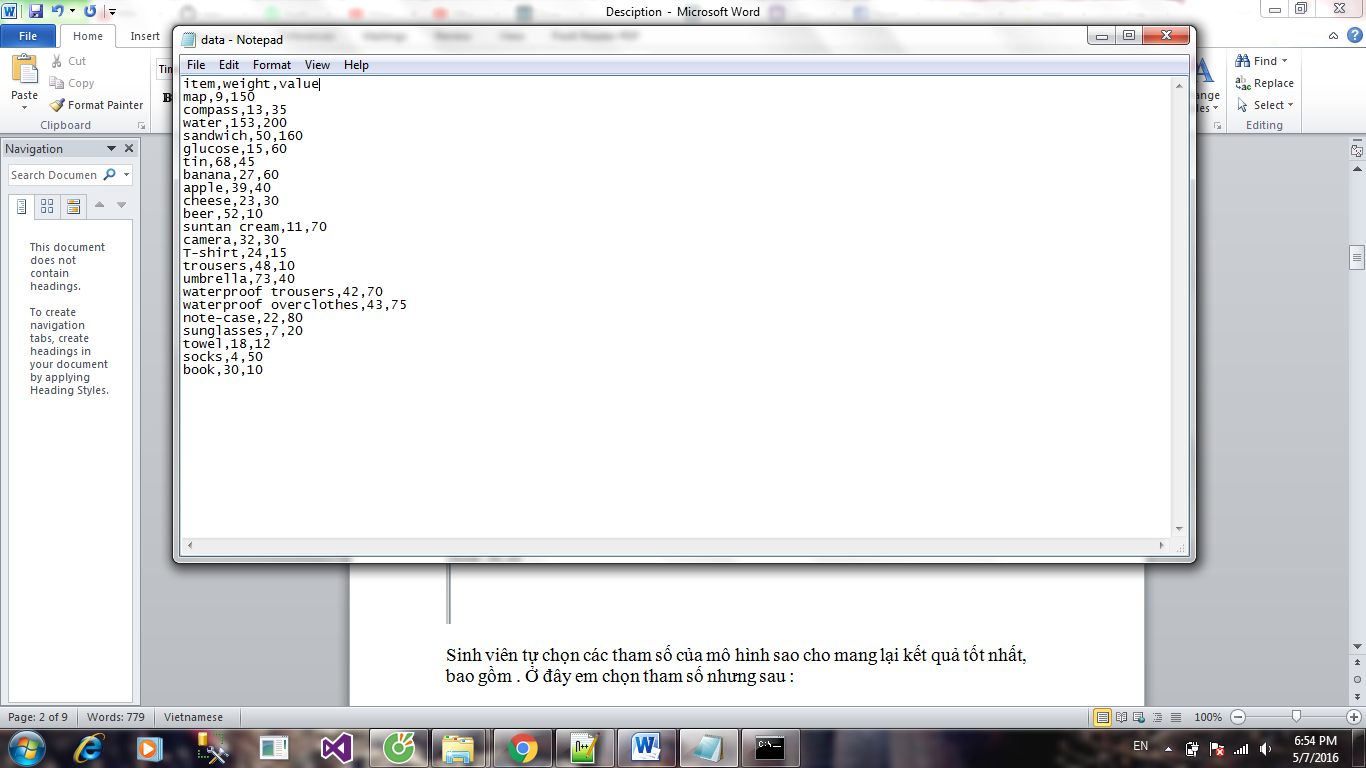
**File dữ liệu đầu vào :**

Em chưa hiểu ý thầy(cô) về cách sắp xếp value và weight lắm nên em xin tự định dạng dữ liệu như sau :   
+ ***Dòng đầu tiên :*** là header item,weight,value

+ **Các dòng tiếp theo** : dữ liệu của từng item theo thứ tự item,weight,value

Còn khối lượng tối đa (weight\_limit) thì e khai báo bên trong khai báo code

***item,weight,value***



Sinh viên tự chọn các tham số của mô hình sao cho mang lại kết quả tốt nhất, bao gồm . Ở đây em chọn tham số nhưng sau :

+ Kích thước quần thể : 10

+ Xác suất lai : 0.4

+ Xác suất đột biến : 0.1

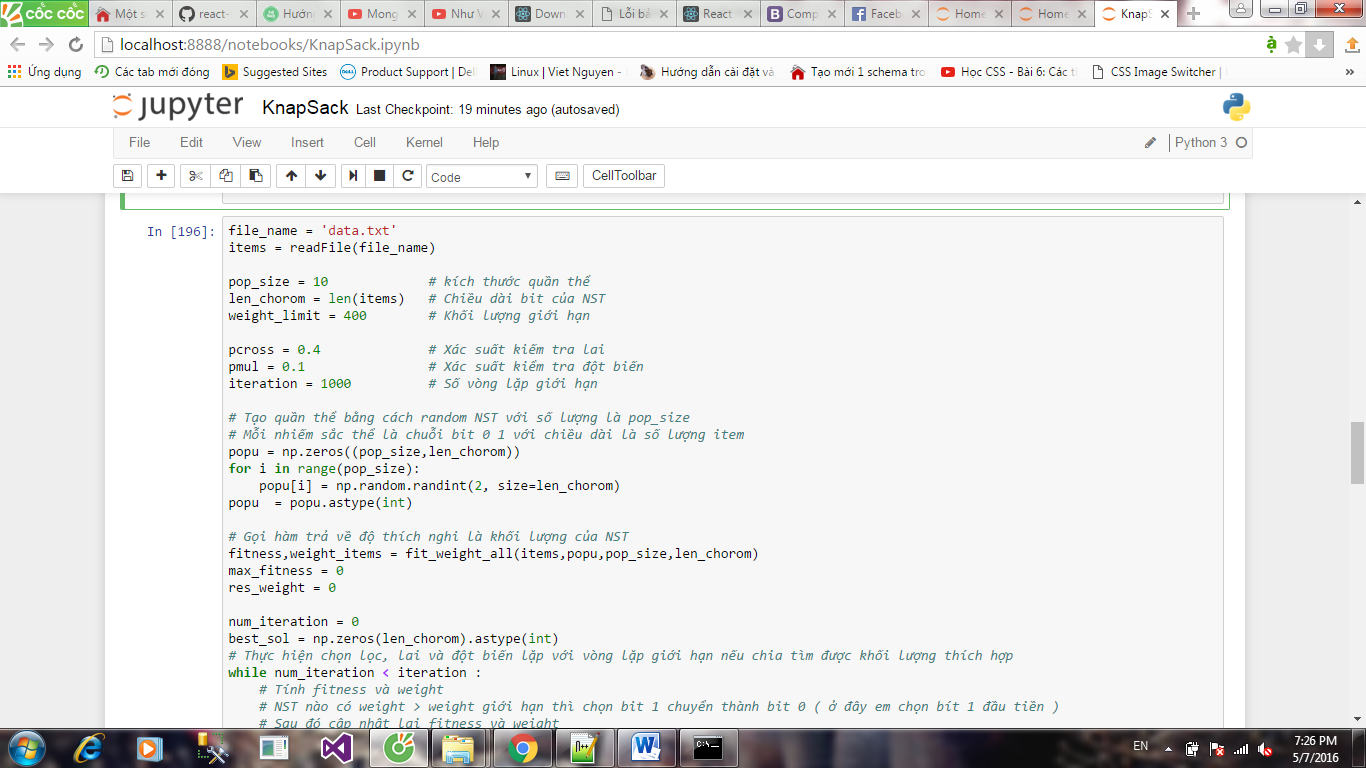
− **Hàm thích nghi**: tổng giá trị các vật được chọn trong túi. Nếu tổng các vật được chọn vượt quá khối lượng tối đa thì chọn bit 1 đầu tiên đảo thành bit 0

**Bài làm**

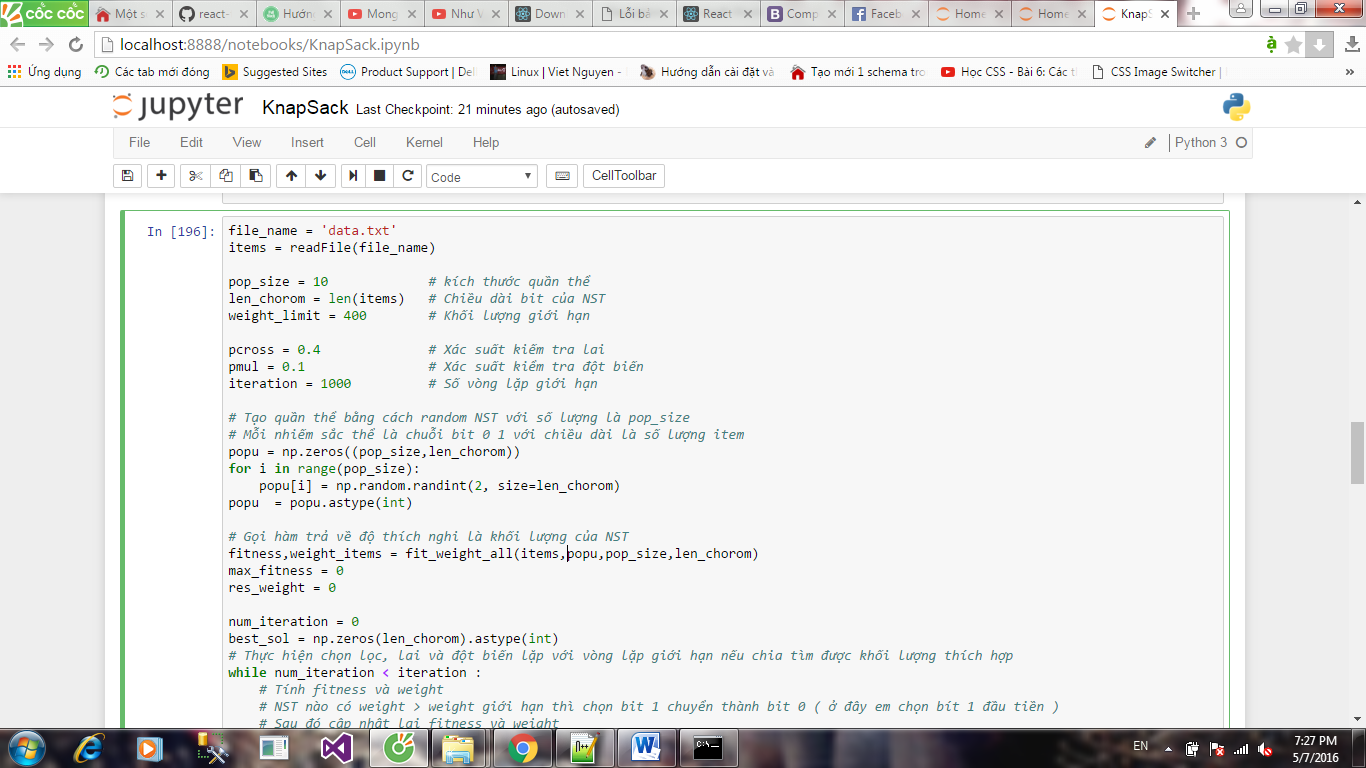
* ***Ngôn ngữ lập trình sử dụng python (2.7)***

**Thực hiện chương trình :**

**+** Đầu tiên thực hiện đọc file : trả về list item . Khai báo các tham số cần thiết



+ Tạo quần thể bằng cách random NST với số lượng là pop\_size . Mỗi nhiếm sắc thể là chuỗi bit 0 1 với chiều dài là số lượng item

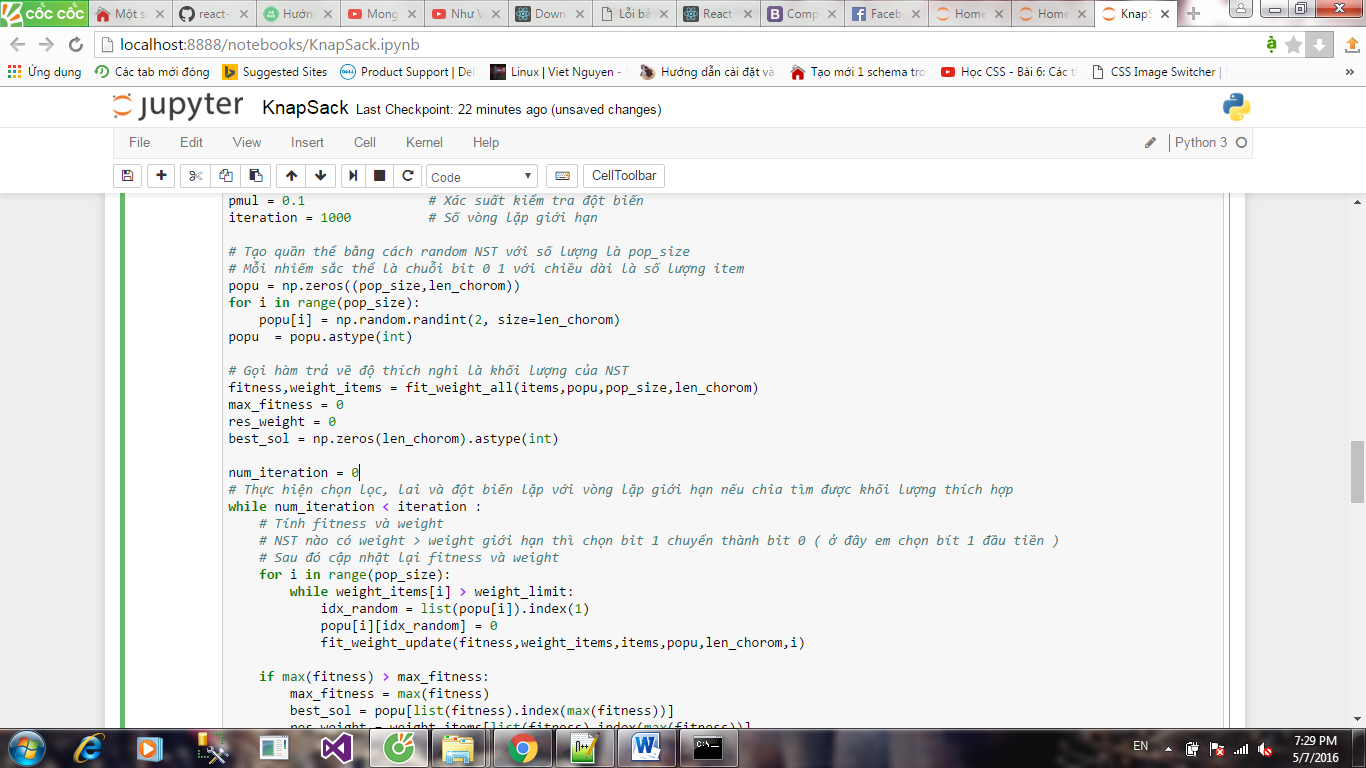


+ Gọi hàm trả về độ thích nghi là khối lượng của NST, khởi tạo những giá trị để lưu NST cho fitness lớn nhất :

max\_fitness : fitness lớn nhất nhất ,

res\_weight : trọng lượng tương ứng với max\_fitness ,

best\_sol : NST có fitness lớn nhất

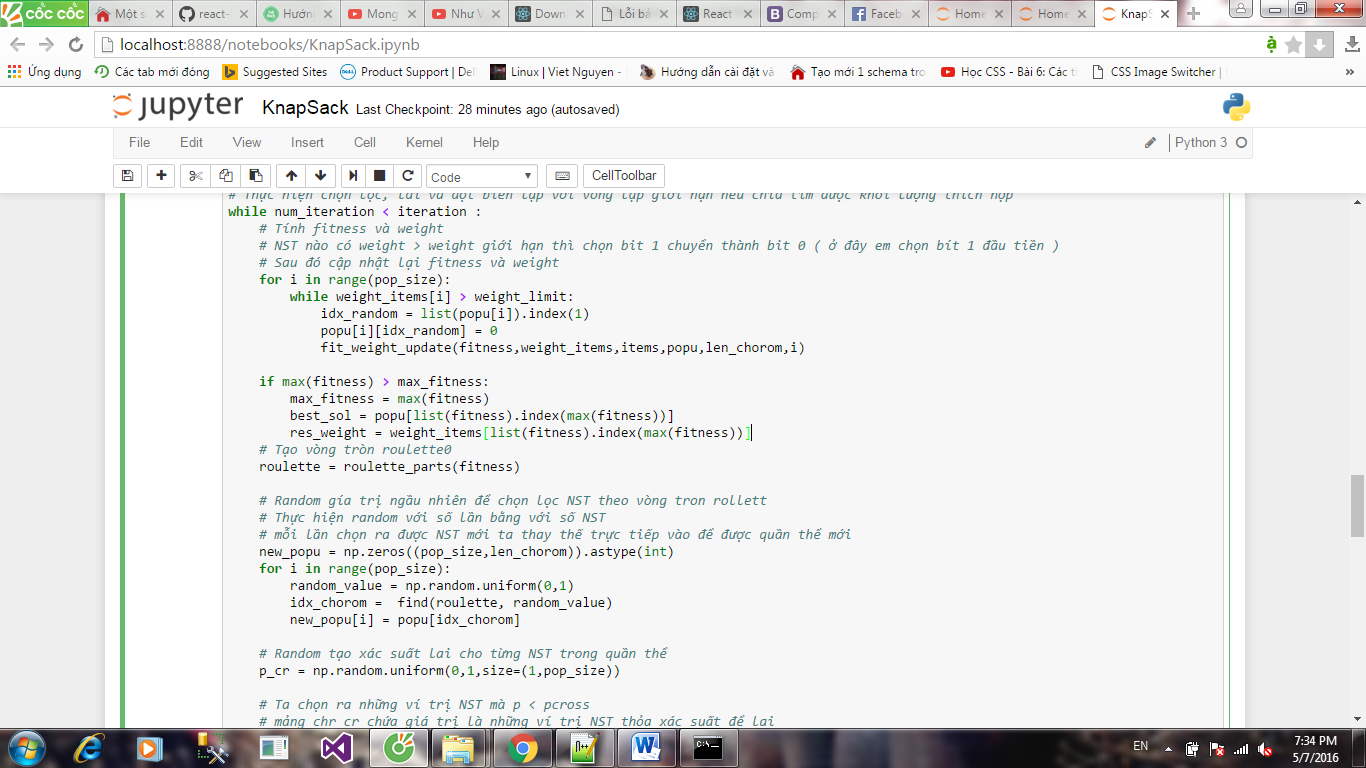


+ Thực hiện vòng lặp :

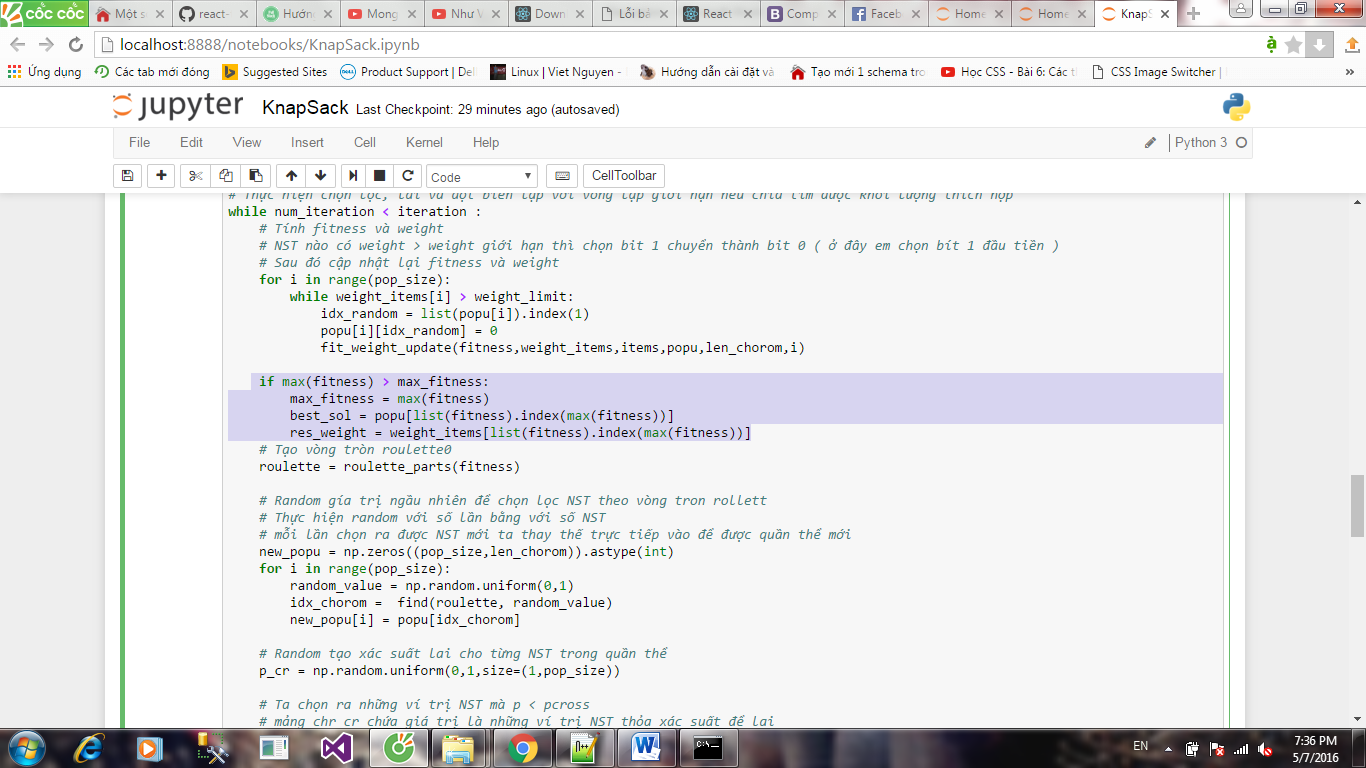
* Tính fitness và weight

NST nào có weight > weight giới hạn thì chọn bit 1 chuyển thành bit 0 ( ở đây em chọn bít 1 đầu tiền )

Sau đó cập nhật lại fitness và weight

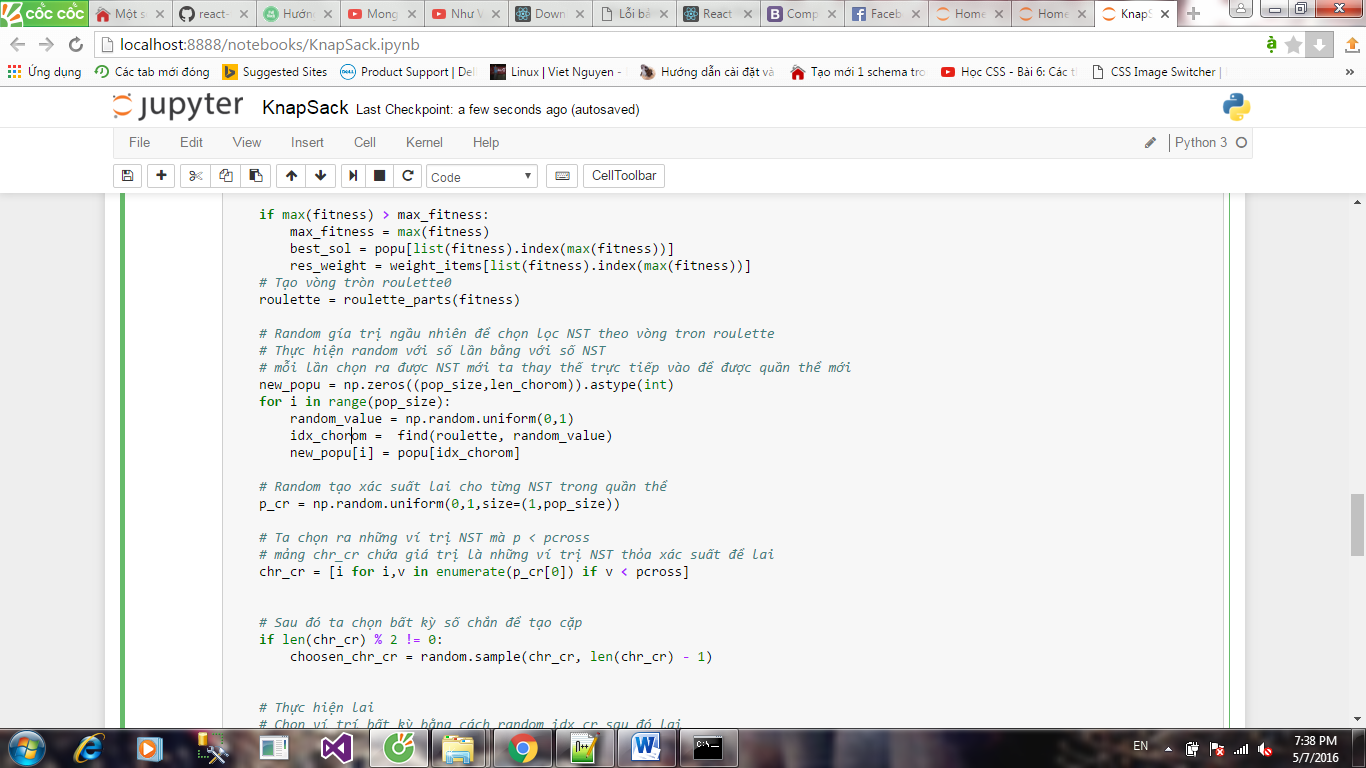


+ Kiểm tra xem độ thích nghi tốt nhất của NST trong quần thể :

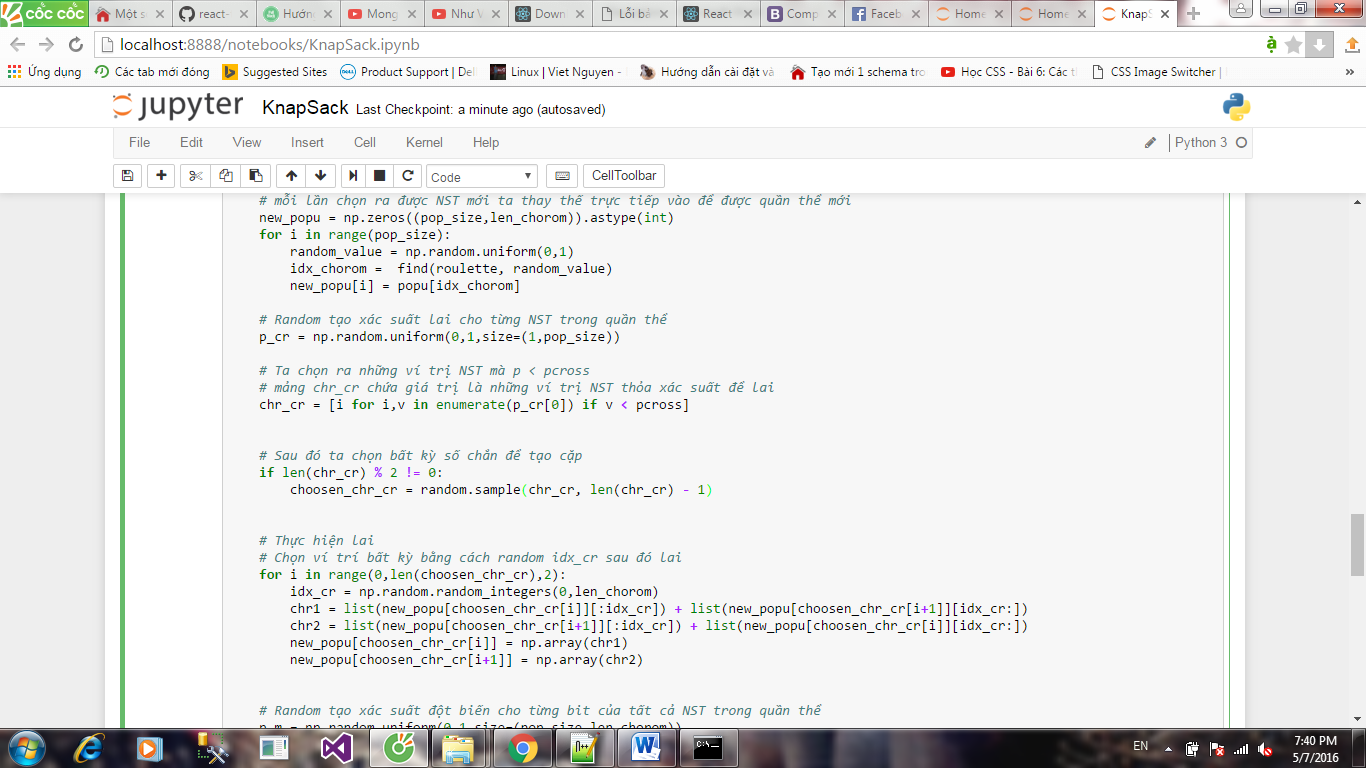


Sau đó tạo vòng tròn roulette bằng cách gọi hàm

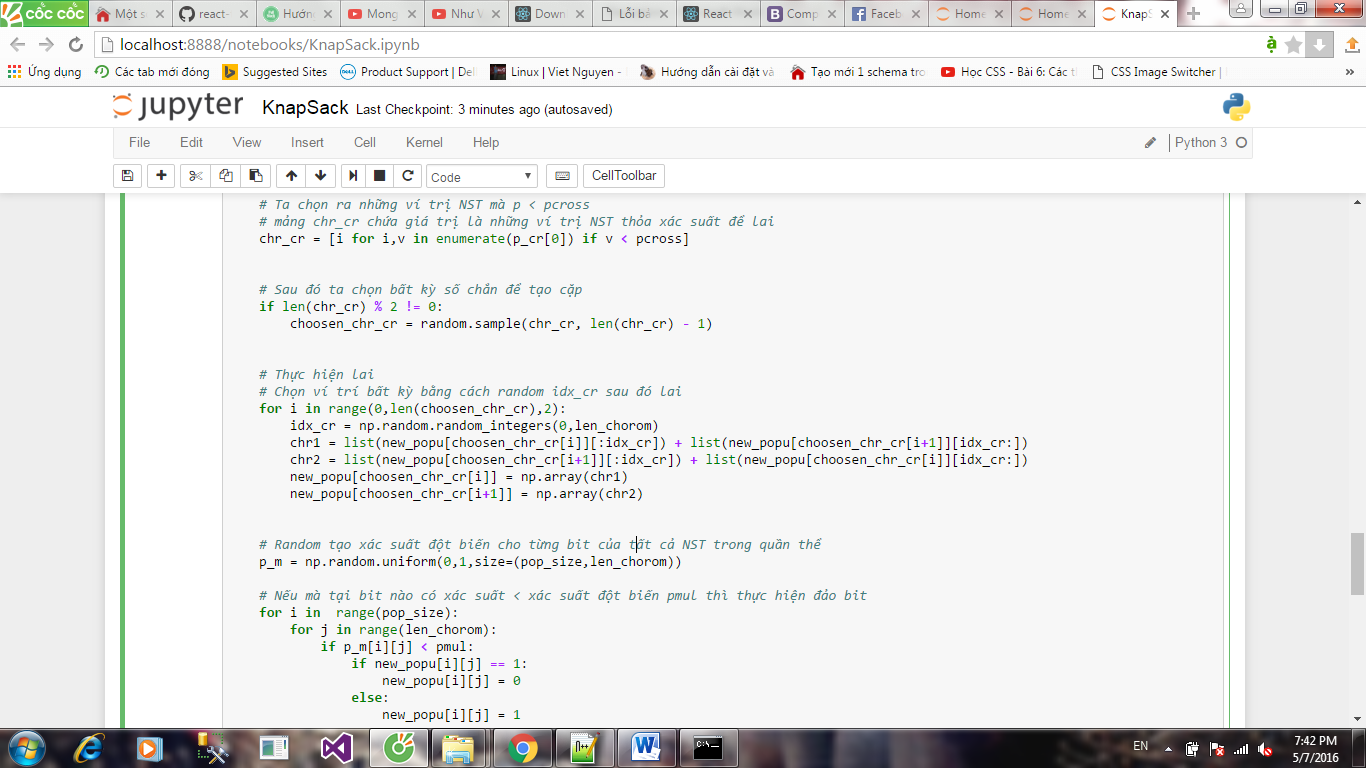
+ Random gía trị ngầu nhiên để chọn lọc NST theo vòng tron roulette Thực hiện random với số lần bằng với số NST, mỗi lần chọn ra được NST mới ta thay thế trực tiếp vào để được quần thể mới



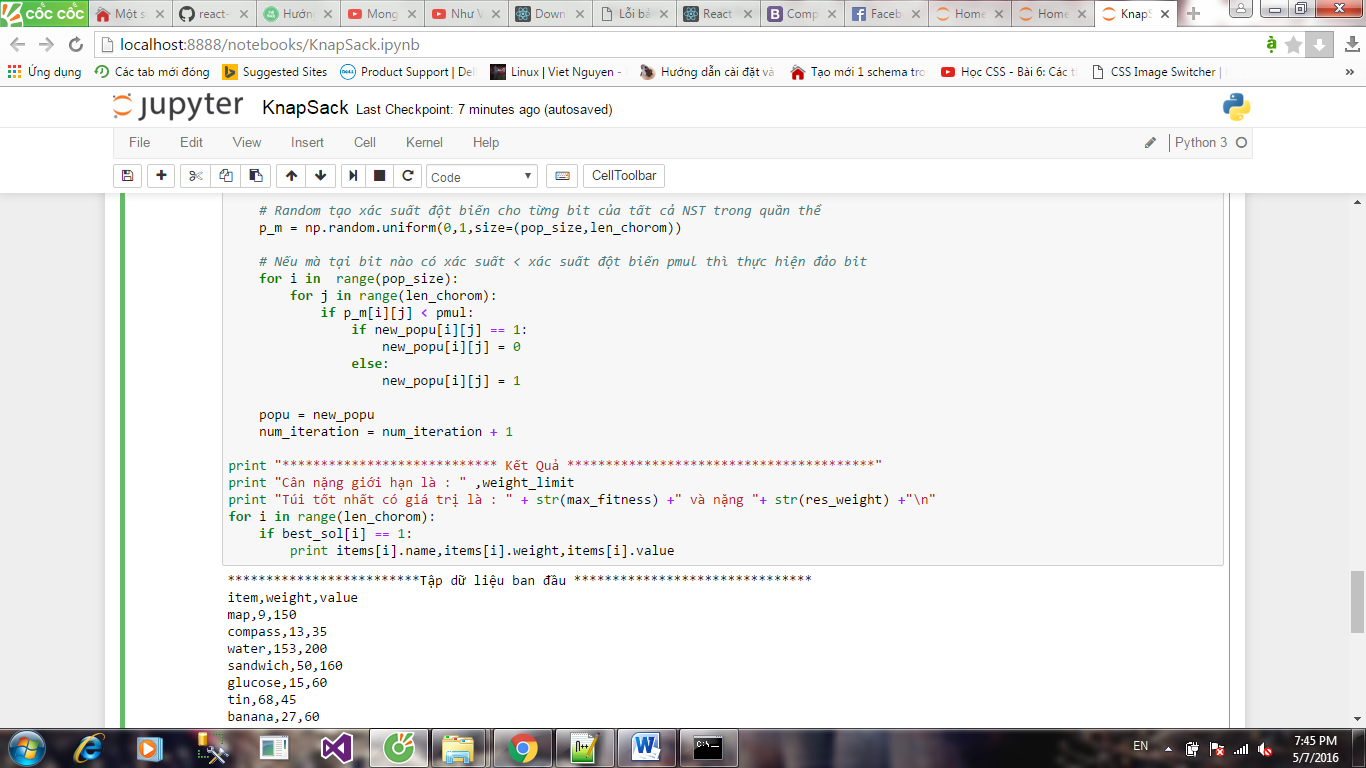
+ Random tạo xác suất lai cho từng NST trong quần thể sau đó Ta chọn ra những ví trị NST mà p < pcross. Sau đó ta chọn bất kỳ số chắn để tạo cặp



+ Thực hiện lai . Chọn ví trí bất kỳ bằng cách random idx\_cr sau đó lai :



+ Random tạo xác suất đột biến cho từng bit của tất cả NST trong quần thể . Nếu mà tại bit nào có xác suất < xác suất đột biến pmul thì thực hiện đảo bit



Sau đó cập nhật quần thể và thực hiện vòng lặp mới

* *Qua nhiều lần chạy kết quả tốt em thu được :*

