**Tài liệu mô tả**

(January 22, 2022)

Sinh viên thực hiện: Đỗ Minh Vũ

Giảng viên hướng dẫn: Th.S Nguyễn Đức Tiến

Project: Subject colector in hust

1. **Giới thiệu**

* Vấn đề đăng ký môn học của sinh viên hiện nay còn nhiều khó khăn. Sinh viên muốn thống kê các môn học trong trường hoặc muốn đăng ký môn nhưng lại không biết hết các môn học điều kiện hay muốn xem thông tin chi tiết các môn học điều kiện. Đó là các vấn đề mà sinh viên có thể gặp phải khi tìm hiểu, đăng ký học tại trường Đại học Bách khoa Hà Nội.
* Project này ra đời để giúp các bạn sinh viên có thể dễ dàng tra cứu cũng như tìm kiếm thông tin một học một cách dễ dàng.

1. **Mục đích**

* Lấy thông tin toàn bộ các môn học trong danh sách môn học tại trường Bách khoa Hà Nội.
* Lấy thông tin một môn học nhất định nào đó.
* Vẽ đồ thị môn học điều kiện.

1. **Nguyên lý**

* Với phần crawl dữ liệu
* Do dữ liệu các môn học ở nhiều trang khác nhau, tuy nhiên url lại không thay đổi => Cần sử dụng selenium để giả lập các thao tác người dùng với web.
* Có 2 trường hợp cần xử lý:
* Nếu chỉ cần tìm thông tin của 1 môn học => xử lý phần lấy mã môn học từ người dùng và nhập vào khung tìm kiếm, sau đó ấn tìm kiếm trên web để hiển thị kết quả tìm được. Sau đó sẽ crawl tất cả dữ liệu này.

try:  
 if (self.code != ""):  
 '''  
 Search by course code  
 '''  
 code\_search = driver.find\_element(By.XPATH, "//\*[@id='MainContent\_tbCourseID\_I']")  
 code\_search.send\_keys(self.code)  
 code\_search.send\_keys(Keys.ENTER)  
 time.sleep(0.5)  
except:  
 logging.info("No argument!!!")

* Nếu người dùng không nhập mã môn học => crawl toàn bộ dữ liệu các môn học trong danh sách.
* Tuy có 2 trường hợp nhưng việc xử lý có những điểm chung: luôn crawl dữ liệu hiển thị trên màn hình. Do đó chỉ cần check xem người dùng có nhập mã môn học hay không, phần còn lại sẽ xử lý như nhau.
* Hàm xử lý cả 2 công việc này trong source là hàm:

def parse\_sourse(self, response):

* Nếu người dùng nhập mã môn học, sử dụng selenium giả lập ấn nút tìm kiếm và nhập dữ liệu người dùng vào, sau đó giả lập ấn “enter” để thực hiện tìm kiếm.
* Với mỗi trang hiển thị, số lượng môn học hiển thị sẽ là 20 (trang cuối có thể ít hơn do không còn đủ 20 môn học).
* Việc crawl từng page hiển thị tiến hành như sau:
* Đếm số lượng môn học hiển thị trong trang bằng cách đếm số dòng (thuộc tính html là tr)
* Sử dụng vòng lặp for để crawl dữ liệu từng môn trong page. Việc đầu tiên là sử dụng selenium giả lập việc ấn nút “+” để hiển thị thông tin mở rộng của môn học. Sau đó crawl tất cả các thông tin của môn học này và lưu vào file csv.
* Khi crawl xong môn cuối cùng của page, sử dụng selenium để giả lập ấn phím điều hướng sang trang tiếp theo (nếu nút next còn hiển thị trên màn hình tức là còn trang tiếp theo và ngược lại).
* Công việc sẽ lặp đi lặp lại cho tới môn học cuối cùng của page cuối cùng.
* Như vậy, tất cả thông tin cần thiết đã được crawl và lưu vào file dưới dạng csv.
* Kết quả như sau:

Text

Description automatically generated

* Với phần vẽ đồ thị
* Sử dụng graphviz để vẽ đồ thị quan hệ của các môn học.
* Người dùng sẽ nhập mã môn học cần tìm kiếm và chương trình sẽ sinh ra đồ thị các môn học điều kiện của môn học đó theo dạng đồ thị cây. Nếu
* Việc xử lý sẽ thực hiện như sau:
* Đọc dữ liệu trong file csv đã crawl được ở trên
* Lưu dữ liệu vào các mảng để truy vấn sau này
* Check xem môn học có môn học điều kiện hay không. Nếu không, chỉ cần vẽ môn học đó ra. Nếu có môn học điều kiện, cần lần lượt tìm các môn học điều kiện tiếp theo bằng cách sử dụng vòng lặp.

for row in data:  
 if(row['Mã HP'] != mahp):  
 continue  
 if(row['Học phần điều kiện']):  
 check = 1  
 hpdk = row['Học phần điều kiện']  
 hpdk = re.sub(r"[^a-zA-Z0-9(),/]", "", hpdk)  
 break  
# if subject has no conditional subjects  
if (check == 0):  
 dot.node(mahp)  
# if subject has conditional subjects

* Sau khi tìm được môn học điều kiện, ta sẽ sử dụng nó làm môn học cần tìm môn học điều kiện. Quá trình tìm kiếm sẽ lặp lại cho tới khi không còn môn học điều kiện nào được tìm ra nữa. Như vậy ta đã tìm được các môn học điều kiện theo chiều sâu của cây.

while(hpdk != ''):  
 dot.edge(mahp, hpdk)  
 mahp = hpdk  
 check = 0  
 for row in data:  
 if (row['Mã HP'] == mahp):  
 hpdk = row['Học phần điều kiện']  
 check = 1  
 break  
 if(check == 0):  
 hpdk = ''

* Kết quả thu được có dạng như sau:



1. **Hạn chế**

* Thời gian crawl dữ liệu tốn nhiều thời gian, có thể xảy ra lỗi nếu mạng sử dụng có tốc độ chậm.
* Phần vẽ đồ thị chưa hiển thị được tất cả các môn học điều kiện theo chiều sâu của cây nếu có nhiều môn học điều kiện.

1. **Tài liệu tham khảo**

* <https://www.datacamp.com/community/tutorials/documenting-python-code>
* <https://graphviz.readthedocs.io/en/stable/manual.html>
* <https://selenium-python.readthedocs.io/>