**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**TOÁN ỨNG DỤNG THỐNG KÊ**

**BÁO CÁO THỰC HÀNH**

**LAB 01**

Mã số sinh viên: 21120582

Họ và Tên: Đinh Hoàng Trung.

Mail: [21120582@student.hcmus.edu.vn](mailto:21120582@student.hcmus.edu.vn).

1. **Khái quát bài làm:**

* Bài tập được làm trên Visual Studio Code.
* Ma trận đầu vào sẽ được nhập vào file .txt.
  + Ví dụ:

A picture containing text

Description automatically generated

Text

Description automatically generated with low confidence

* Ma trận sau khi được giải nghiệm sẽ xuất ra màn hình terminal (VSC) / console.
* Thư viện được sử dụng : numpy.
* Các hàm theo yêu cầu bài tập:
  + innerproduct(v1, v2): trả về tích vô hướng của 2 vector v1 và v2 dưới dạng số thực.
  + QR\_factorization(A): Truyền vào 1 ma trận A bất kì khác 0 trả về ma trận phân rã của ma trận A dưới dạng A = QR
* Các hàm hỗ trợ:
  + print\_matrix\_list<ma trận\_dạng list>: in ra ma trận lưu trong list.
  + multiplyScalarVector< biến số thực hoặc nguyên, vector>: hàm trả về kết quả của vector nhân 1 hằng số.
  + transpose<ma trận>: trả về ma trận chuyển vị.
  + round\_n <ma trận, số nguyên>: làm tròn các giá trị trong ma trận với số thập phân muốn làm tròn được truyền vào.
  + nhanmatran <ma trận A, ma trận B>: trả về kết quả là list lưu ma trận kết quả của phép nhân 2 ma trận A, B.
  + module<vector>: trả về độ dài của vecrtor truyền vào.
  + xuli0<ma trận A>: đưa các giá trị xấp xỉ 0 của ma trận về giá trị 0 để hỗ trợ cho việc in ra ma trận (dễ nhìn hơn).
  + GramSchmidtProcess<ma trận A> : trả về ma trận trực giao gồm các vector được trực chuẩn hóa (Q).

1. **innerproduct( v1, v2 ).**
   1. Kiểm tra điều kiện đủ để thực hiện phép tính:

* Kiểm tra chiều dài của 2 vector.
* Nếu khác nhau thì in ra màn hình cảnh báo và dừng hàm.

Vd:

Text

Description automatically generated

* 1. Nếu thỏa điều kiện ( độ dài 2 vector bằng nhau ).

Thì trả về tích vô hướng của 2 vector.

Vd:



1. **QR\_factoriztion(A).**
   1. Tìm ma trận Q qua hàm GramSchmitdtProcess.

* Giải thuật:

Text, letter

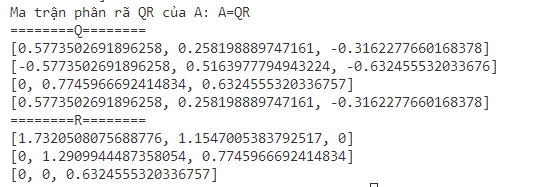
Description automatically generated

* + Lần lượt trực hóa các vector cơ sở của ma trận A.
  + Các vector sau được trực hóa dựa vào các vector trước.
  + Sau khi trực hóa tất các các vector thì chuẩn hóa toàn bộ các vector vừa trực hóa.
* Note: có thể vừa trực hóa vừa chuẩn hóa thay vì trực hóa hết tất cả các vector trong ma trận nhưng 2 cách làm khác nhau sẽ đưa ra kết quả khác nhau, tuy nhiên theo em cả 2 đều không sai vì đưa ra được các vector cơ sở trực giao và có độ dài bằng 1. (ở bài này em trực hóa hết cả ma trận rồi mới chuẩn hóa)
* Ví dụ:
  + Với ma trận Input là:

Text

Description automatically generated with low confidence

* + Với cách vừa trực hóa vừa chuấn hóa các vector ta thu được ma trận QR: (cách 1)(ý tưởng ban đầu)



Chuẩn hóa các vector trực giao ta thu được (cách 2): (đã được làm tròn 2 chữ số thập phân)

Text

Description automatically generated

* Ý tưởng ban đầu là cách 1:
  + Theo kết quả tra được trên phần mềm thì vẫn đúng nhưng theo giải thuật được học trên lớp là cách 2 và theo kết quả giải tay thì được kết quả đúng với cách 2:
    - Kết quả trên phần mềm:

A picture containing table

Description automatically generated

* + - Kết quả giải tay trên lớp:

A picture containing text, whiteboard, tiled

Description automatically generatedA picture containing text, whiteboard

Description automatically generated

* Q là ma trận trực giao của ma trận góc.
  + (kết quả đã được làm tròn đến 2 chữ số thập phân)

A picture containing letter

Description automatically generated

* 1. Khi có Q ta tìm R qua công thức Q­­-1A.
* Q-1 = QT.

A picture containing shape

Description automatically generated

----Hết---