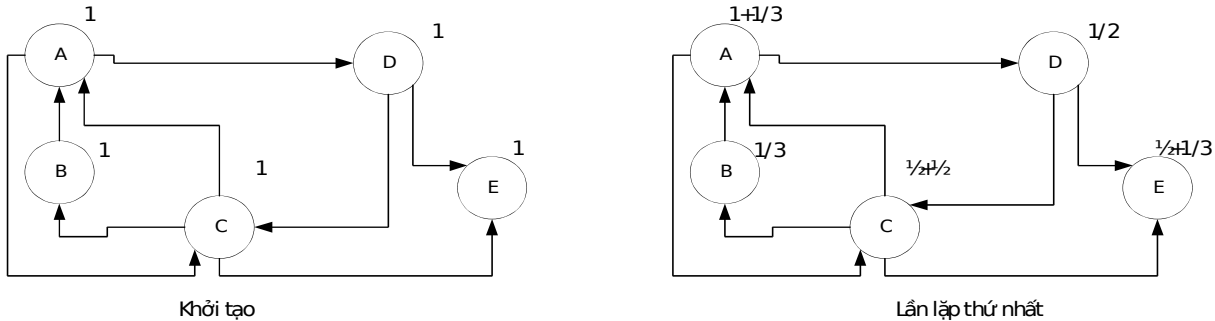


## Đề thi Lập trình C nâng cao

(Thời gian làm bài 90', được sử dụng tài liệu, đề thi gồm 2 mặt giấy)

Trong mạng internet, mỗi trang web (webpage) đều có các liên kết đến và đi tới các trang web khác. Để đánh giá mức độ quan trọng của các webpage, Google đã đề xuất một thuật toán **PageRank** khá đơn giản như sau.



Khởi tạo, mỗi webpage sẽ được gán điểm quan trọng là 1. Trong mỗi lần lặp, điểm của mỗi webpage sẽ được cập nhật bởi giá trị điểm của các webpage trỏ tới nó trong lần lặp ngay trước.

Ví dụ với hình minh họa ở trên, Page A được trỏ tới bởi B và C, PageRank của A sẽ là  $PR(A) = PR(B)/3 + PR(C) = 1/3 + 1$

Vì C có 3 liên kết đi ra, nên trọng số mỗi liên kết này là  $PR(C)/3$ , còn B chỉ 1 liên kết đi ra và tới A nên trọng số liên kết này là  $PR(B)$ . Tương tự, ta có thể tính được trọng số của các nút còn lại trong mạng trong lần lặp thứ nhất.

Với lần lặp thứ 2 công thức vẫn tương tự, tuy nhiên ta sẽ sử dụng giá trị là trọng số của các webpage trong lần lặp 1.

Cho đầu vào là 2 file văn bản:

- Tập **webpages.txt** chứa url của các webpage và id của tương ứng là số nguyên. Dòng 1 là số lượng n webpage; n dòng tiếp theo lần lượt là url và id (được cách nhau bởi 1 dấu cách trống) của từng webpage. Url là 1 chuỗi ký tự (chỉ gồm các mã ASCII) có độ dài tối đa 250 và không có dấu cách trống. Ví dụ:

```
=====
4
https://www.hust.edu.vn/gioi-thieu 1012
https://www.hust.edu.vn/thong-diep-cua-hieu-truong 1010
https://www.hust.edu.vn/thong-tin-tuyen-sinh 1013
https://www.hust.edu.vn/tuyen-sinh-nghien-cuu-sinh 1015
=====
```

- Tập **pageConnections.txt** chứa các liên kết giữa các webpage. Dòng 1 là số lượng n webpage có trong file n dòng tiếp theo là id webpage hiện tại và danh sách các id được trỏ tới từ webpage hiện tại, mỗi id được ngăn cách bởi 1 dấu cách trống. Ví dụ:

```

=====
4
1012 1010
1010 1012
1013 1015 1012
1015 1013 1012 1010
=====

```

Yêu cầu viết một chương trình (được sử dụng cấu trúc dữ liệu tùy ý để lưu trữ dữ liệu đồ thị) có các chức năng sau:

- A. In ra một menu gồm 6 số (5 số đầu tiên có chức năng như mô tả ở dưới), số 6 là Thoát (thoát chương trình) (1đ).  
 Chú ý: Với 5 số ứng với 5 chức năng ở dưới, sau khi thực hiện xong từng chức năng thì chương trình sẽ tự động quay lại menu chính.
- B. Từ menu chính, ấn số 1: đọc vào nội dung 2 file trên:
  1. In ra số lượng webpage và url của từng webpage kèm theo id tương ứng (ngăn với url bởi dấu cách trống) (2đ)
  2. In ra webpage nào có nhiều liên kết tới nhất (1đ) và ít liên kết tới nhất (1đ)
- C. Từ menu chính, ấn số 2: thực hiện thuật toán **PageRank** với khởi tạo trọng số của các page ban đầu bằng 1. Sau lần lặp đầu tiên, in ra màn hình url đầy đủ của webpage nào có trọng số cao nhất (1đ) và webpage nào có trọng số thấp nhất (1đ).
- D. Từ menu chính, ấn số 3: Thực hiện thuật toán **PageRank** sau  $m$  lần lặp (với  $m$  được nhập từ bàn phím và có giá trị trong khoảng 10-50) và in ra màn hình top 3 webpage quan trọng nhất. (1đ)
- E. Từ menu chính, ấn số 4:
  1. In ra số lượng webpage trong hệ thống mà chỉ có liên kết đến, nhưng không có liên kết ra ngoài (0.5đ).
  2. In ra số lượng các spam bot có trong hệ thống (các webpage chỉ có liên kết ra mà không có liên kết đến) (0.5đ).
- F. Từ menu chính, ấn số 5: Nhập từ màn hình 2 số nguyên là  $id1$  và  $id2$ . In ra khoảng cách nhỏ nhất từ webpage có  $id1$  tới webpage có  $id2$ . Kết quả in ra sẽ là -1 nếu  $id1$  bằng  $id2$ , hoặc  $id1$  ( $id2$ ) không hợp lệ, hoặc không tồn tại trong dữ liệu đọc vào, hoặc không tồn tại đường đi từ webpage có  $id1$  tới webpage có  $id2$  (1đ). **Chú ý:** khoảng cách nhỏ nhất giữa 2 webpage không trùng nhau là số cạnh đếm được trên đường đi ngắn nhất từ webpage có  $id1$  tới webpage có  $id2$  (coi tất cả các cạnh có trọng số bằng 1). Ví dụ: khoảng cách nhỏ nhất từ webpage C tới webpage A là 1.