PageRank

1. Giới thiệu PageRank:

- PageRank là thuật toán được công bố bởi Larry Page (Google) vào năm 1998.
- Mục tiêu: Xác định mức độ **quan trọng** của một trang web dựa trên **liên kết (links)** đến và đi từ các trang khác.

2. Mô hình hóa mạng web:

- Mỗi trang web được đại diện bởi một **nút (node)** trong mạng.
- **Procrastinating Pat** – một người dùng tưởng tượng nhấp ngẫu nhiên vào các liên kết, được dùng để mô phỏng hành vi duyệt web.
- Mỗi trang có một **vector liên kết (link vector)**, được chuẩn hóa để thể hiện **xác suất** nhấp vào liên kết từ trang đó.

3. Ma trận liên kết (L):

- Mỗi cột của **ma trận L** là một vector liên kết từ một trang.
- Mỗi phần tử L_{ij} đại diện xác suất từ trang **j** chuyển sang **i**.
- Đặt **r** là vector thứ hạng các trang, ta có phương trình:

$$r = Lr$$

- Đây là một **eigenvector** của ma trận **L** với **eigenvalue = 1**.

4. Giải bài toán bằng phương pháp lặp (Power Method):

- Bắt đầu từ một vector r đều (ví dụ: [0.25, 0.25, 0.25, 0.25]).
- Lặp: $r_{i+1} = Lr_i$ cho đến khi r hội tụ (không thay đổi nữa).
- Đây là **Power Method**, dùng để tìm eigenvector tương ứng với eigenvalue lớn nhất (ở đây là 1).

5. Kết quả ví dụ:

- Sau khoảng 10 lần lặp, vector r hội tụ:
  - Trang D chiếm ~40% thời gian của người dùng (quan trọng nhất),
  - A thấp nhất (~12%),

- B và C bằng nhau (~24%).

6. Sparse Matrix & Tính hiệu quả:

- Trong thực tế, ma trận liên kết rất **thưa (sparse)** vì hầu hết trang web không liên kết đến nhau.
- Nhân ma trận sparse rất hiệu quả, giúp PageRank mở rộng cho hàng **tỷ** trang.

7. Hệ số suy giảm (Damping Factor):

- Được thêm vào công thức để cải thiện tính ổn định:

$$r_{i+1} = dLr_i + \frac{1-d}{n}$$

  - d thường là 0.85.
  - Phản ánh khả năng người dùng nhập URL trực tiếp thay vì nhấp link.