Comments. The fact that $\vec{p}$ is a vector of PageRanks means that $\vec{p}$ is the only non-trivial vector such that

$$\left(\lambda M^T + (1-\lambda)\frac{J}{n}\right)\vec{p} = \vec{p}.$$

The fact that that the previous equation holds (with $\lambda < 1$) does not imply, tentatively, that $M^T\vec{p} = \vec{p}$ also holds. We know that if M is stochastic, there is at least one vector $\vec{p'}$ that satisfies $M^T\vec{p'} = \vec{p'}$ but we do not know if $\vec{p}$, as defined above, is one of these vectors (i.e. a vector that satisfies $M^T\vec{p} = \vec{p}$). We have to prove it.

Exercici 4

(1 punt) We have designed a sophisticated recommender system based on collaborative filtering and we would like to evaluate its performance by comparing it against a simple system that does not employ collaborative filtering. We wish to make sure that the sophistication of our system is worth.

1. What system would you use for comparison?
2. Why?

Solució de l'exercici 4

1. The baseline would recommend the most frequently selected items.
2. Because that neglects any information about user past choices.

Exercici 5

(2 punts) Consider the set of edges of a large undirected graph consisting of unordered pairs of the form (u, v), where u and v are integers that stand for two distinct vertices. In this setting, we wish to solve various problems that take the set of edges as input applying the MapReduce programming paradigm.

1. Calculate the degree of every vertex using just one job. The output of the *reduce* functions must be pairs of the form (v, k), where k is the degree of v.
2. Calculate the so-called *average degree of nearest neighbours* of each vertex. Given a vertex i, such average is

$$k_{nn}(i) = \frac{1}{k_i}\sum_{j=1}^{n} a_{ij}k_j, \tag{4}$$

   where k_j is the degree of the j-th vertex and $a_{ij} = 1$ if vertices i and j are connected (otherwise $a_{ij} = 0$). Split the programming into two jobs

   - The first job has to compute the set of nearest neighbours of a vertex v. The output of the *reduce* functions are pairs of the form (v, Γ), where Γ is the set of vertices adjacent to v.
   - The second job takes the output of the 1st job as input of the *map* functions. The output of the *reduce* functions are pairs of the form $(v, k_{nn}(v))$.

Please provide the pseudocode of *map*, *reduce* (and optionally *combine*) functions. Solutions are expected to be simple and efficient.

Solució de l'exercici 5

1. ```
   map(u, v)
     output (u, 1)
     output (v, 1)

   reduce(v, L) = combine(v, L)
     output (v, sum(L))
   ```
   Solutions without *combine* function are less efficient.
   ```

2. First job:

```
map(u, v)
  output (u, v)
  output (v, u)

reduce(v, L)
  output (v, L)
```

Second job (G is the set of vertices adjacent to a certain vertex):

```
map(v, G)
  for each vertex u in G
     output (u, size of G)

combine(v, L)
  output (u, [sum(L), size(L)])

reduce(v, L)
  s is the sum of 1st elements of pairs in L
  k is the sum of 2nd elements of pairs in L // k is actually the degree of v
  output (v, s/k)
```

A less efficient version without *combine*

```
map(v, G)
  for each vertex u in G
     output (u, size of G)

reduce(v, L)
  output (v, sum(L)/size(L))
```

Exercici 6

(2 punts) The average shortest path length of an undirected graph of n vertices is defined as

$$l = \frac{1}{\binom{n}{2}} \sum_{i<j} d_{ij}. \quad (5)$$

Suppose a Watts-Strogatz model with parameters n (number of vertices), p (rewiring probability) and K (the mean vertex degree).

1. Estimate (approximately) the value of l for $p = 1$ and sufficiently large n?
2. If you wished to use an Erdős-Rényi graph, $G_{n,\pi}$ as a control (or baseline) for a Watts-Strogatz model, which value would you use to set the parameter π?
3. Calculate l exactly as a function of n when $K = 2$ and $p = 0$.

Solució de l'exercici 6

1. When $p = 1$ the Watts-Strogatz model is equivalent to an Erdős-Rényi graph, where $l \approx \frac{\log n}{\log z}$, where z is the average degree. As $z = K$, $l \approx \frac{\log n}{\log K}$.
2. π has to be the density of links, i.e.

$$\pi = \frac{Kn/2}{\binom{n}{2}} = \frac{K}{n-1}.$$

3. Since distances are symmetric ($d_{ij} = d_{ji}$), l can be expressed as

$$l = \frac{1}{n(n-1)} \sum_{i=1}^{n} D_i,$$

where

$$D_i = \sum_{j=1}^{n} d_{ij}.$$

In a regular lattice, D_i is the same for every vertex and then

$$l = \frac{1}{n-1} D_1.$$

When n is odd, minimum vertex-vertex distances range between 1 and $(n-1)/2$,

$$\begin{aligned} D_1 &= 2 \sum_{\delta=1}^{(n-1)/2} \delta \\ &= \frac{1}{4}(n-1)(n+1) \end{aligned}$$

and then

$$l = \frac{n+1}{4}$$

for $n \geq 2$. When n is even, minimum vertex-vertex distances range between 1 and $(n-1)/2$ and

$$\begin{aligned} D_1 &= 2 \sum_{\delta=1}^{n/2-1} \delta + n/2 \\ &= \frac{n^2}{4} \end{aligned}$$

and then

$$l = \frac{n^2}{4(n-1)}$$

for $n \geq 2$. All together,

$$l = \frac{(n-x)(n+x)}{4(n-1)}$$

where $x = 1 - n \bmod 2$.