# Programowanie i metody numeryczne

Ćwiczenia 6.

Zagadnienie własne.

## Zadanie 1. Metoda QR.

a) Napisz program `eigqr` wczytujący elementy macierzy kwadratowej $A$ z pliku `A.txt` i wypisujący na standardowe wyjście wartości i wektory własne tej macierzy oraz odpowiednie oszacowanie błędu. Plik `A.txt` powinien zawierać $N$ linii, a w każdej z nich $N$ liczb oddzielonych. Posłuż się własną implementacją metody QR.

Przetestuj działanie programu dla różnych macierzy, których wektory i wartości jesteś w stanie wyznaczyć samodzielnie.

b) Napisz program `randomqr` przyjmujący jako argument wywołania liczbę naturalną $K$. Program powinien tworzyć losowe macierze $k \times k$ dla $k = 1, \ldots, K$, a następnie wyznaczać ich wektory i wartości własne korzystając z Twojej implementacji metody QR oraz odpowiedniej funkcji bibliotecznej. Wynikiem działania programu powinien być rysunek przedstawiajacy na jednym wykresie czas wykonywania obliczeń w funkcji $k$.

## Zadanie 2. Algorytm PageRank.

Wyszukiwarki internetowe zwracają wszystkie strony internetowe powiązane z frazą podaną przez użytkownika – przede wszystkim takie, w których tytule lub treści fraza ta występuje. Pozostaje jednak problem uporządkowania wyników wyszukiwania: którą stronę należy wyświetlić jako pierwszy wynik, którą jako drugi etc. Jednym z rozwiązań tego zadania jest algorytm PageRank [1] [2], opracowany przez twórców wyszukiwarki Google, Larry'ego Page'a i Sergeya Brina w 1998 roku, w czasie ich studiów na Uniwersytecie Stanforda.

Przedstawiony poniżej uproszczony opis algorytmu Page Rank został opracowany na podstawie [3] i [4].

PageRank przypisuje każdej stronie internetowej współczynnik opisujący jej rangę, opierając się na liczbie odnośników do tej strony znalezionych na innych stronach. Niech $N$ będzie liczbą wszystkich stron internetowych. Wprowadźmy macierz $A$ opisującą strukturę odnośników pomiędzy stronami – jej elementy macierzowe będą miały postać

$$a_{ij} = \begin{cases} 1, & \text{gdy na stronie } j \text{ jest odnośnik do strony } i, \\ 0, & \text{gdy na stronie } j \text{ nie ma odnośnika do strony } i \text{ lub gdy } i = j. \end{cases}$$

Przyjmujemy dla uproszczenia, że nie bierzemy pod uwagę liczby linków ze strony $j$ do strony $i$ – gdy jest ich więcej, traktujemy je jako jeden odnośnik. Nie bierzemy również pod uwagę odnośników ze strony do niej samej – stąd warunek $a_{ij} = 0$, gdy $i = j$. Spodziewamy się, że macierz $A$ będzie rozrzedzona, czyli że będzie zawierała wiele zer, ponieważ na typowej stronie internetowej nie ma zbyt wielu linków.

Zastosujemy następujący model zachowania internauty:

- jeśli na stronie, na której znajduje się internauta, nie ma ani jednego odnośnika do innej strony, internauta po opuszczeniu tej strony przejdzie do innej, losowo wybranej strony; prawdopodobieństwo przejścia ze strony $j$ do strony $i$ to w tym przypadku $1/N$,

- jeśli na stronie, na której znajduje się internauta, są odnośniki do innych stron, wówczas dla pewnego $\alpha \in [0, 1]$:
  - z prawdopodobieństwem $\alpha$: internauta kliknie jeden z linków na tej stronie; wybór konkretnego linku będzie losowy, zatem prawdopodobieństwo przejścia ze strony $j$ do strony $i$ jest tym razem równe $a_{ij}/d_j$, gdzie $d_j = \sum_i a_{ij}$ to liczba wszystkich linków na stronie $j$,
  - z prawdopodobieństwem $1-\alpha$: internauta nie kliknie w żaden z linków i przejdzie do innej, losowo wybranej strony; prawdopodobieństwo przejścia ze strony $j$ do strony $i$ to znowu $1/N$.

Tak więc prawdopodobieństwo przejścia internauty ze strony $j$ na stronę $i$ jest równe

$$p_{ij} = \begin{cases} \alpha \dfrac{a_{ij}}{d_j} + \dfrac{1-\alpha}{N}, & \text{gdy } d_j \neq 0, \\ \dfrac{1}{N}, & \text{gdy } d_j = 0. \end{cases}$$

W oryginalnej pracy [1] przyjęto $\alpha = 0,85$.

Załóżmy, że na początku internauta znajdzie się na stronie $i$ z prawdopodobieństwem $x_i^{(0)}$; wektor $x^{(0)}$ o składowych $x_i^{(0)}$ opisuje więc prawdopodobieństwa dla wszystkich stron. Po pierwszym przejściu internauty na inną stronę wektor prawdopodobieństw to $x^{(1)} = Px^{(0)}$, gdzie $P$ jest macierzą o elementach macierzowych $p_{ij}$ etc. Otrzymujemy zależność rekurencyjną: $x^{(k+1)} = Px^{(k)}$. Jeśli istnieje granica $x^\star = \lim_{k\to\infty} x^{(k)}$, musi ona spełniać

$$x^\star = Px^\star,$$

czyli być wektorem własnym macierzy $P$ dla wartości własnej $\lambda = 1$.

Algorytm PageRank przypisuje każdej stronie współczynnik, który jest odpowiednią składową wektora $x^\star$: ranga strony $i$ to $x_i^\star$. Można wykazać, że:

- macierz $P$ ma wartość własną równą 1,
- każda wartość własna $\lambda$ macierzy $P$ spełnia $|\lambda| \leq 1$,
- 1 jest jedyną wartością własną $P$ o module równym 1.

W celu wyznaczenia wektora $x^\star$ możemy więc użyć podstawowej wersji metody potęgowej dla dominującej wartości własnej.

a) Napisz funkcję `powereig`, która dla zadanej macierzy zwraca wektor własny odpowiadający dominującej wartości własnej oraz tę wartość własną.

b) Napisz program `pagerank`, który wczyta dane z pliku `in.txt`, opisującego przykładowe zależności pomiędzy kilkoma hipotetycznymi stronami internetowymi, a następnie wypisze ranking tych stron wraz z ich rangami.

c) *DLA CHĘTNYCH.* Napisz program `wikirank`, przeanalizuje wszystkie strony polskiej Wikipedii i na podstawie odnośników na tych stronach wykona ich ranking, wypisując 100 najpopularniejszych stron na polskiej Wikipedii oraz ich rangi.

*UWAGA.* Program rozwiązujący to zadanie może wykonywać się długo.

*Opracowanie: Bartłomiej Zglinicki.*

## Literatura


[1] Brin S., Page L., *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Computer Networks **30** (1998) 107-117, `http://infolab.stanford.edu/~backrub/google.html`

[2] Brin S., Motwani R., Page L., Winograd T., *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford InfoLab 1999, `http://ilpubs.stanford.edu:8090/422/`

[3] Krzyżanowski P., *Metody numeryczne*, Wydawnictwo Naukowe PWN, Warszawa 2024.

[4] Austin D., *How Google Finds Your Needle in the Web's Haystack*, w: *The Feature Column*, American Mathematical Society, 2006, `https://www.ams.org/publicoutreach/feature-column/fcarc-pagerank`