

PageRank-algorithmus

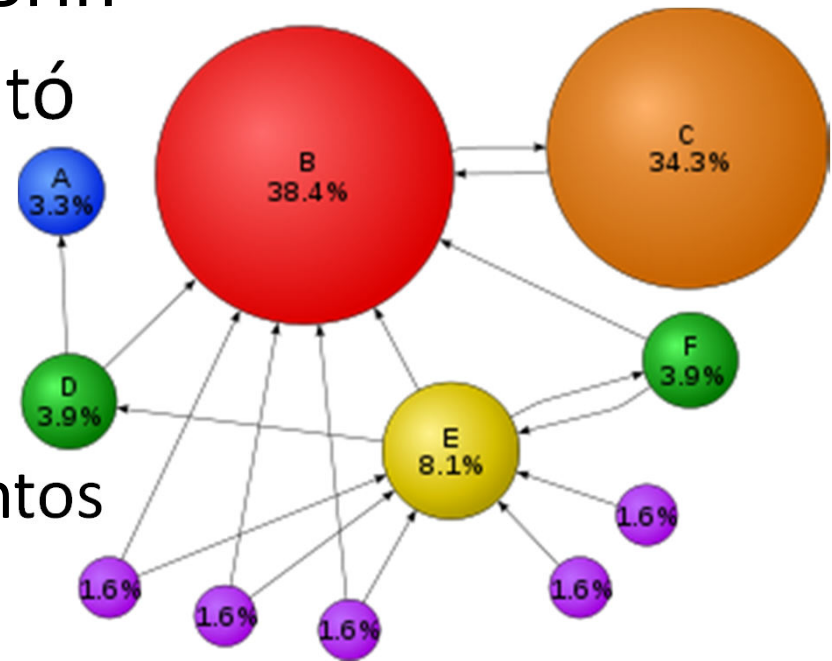
The image shows a screenshot of a BBC Four video player. The video content features a man in a red jacket speaking on a golf course. A search bar is overlaid on the video, displaying the search term 'pagerank' and the results: 'About 31,800,000 results (0.19 seconds)'. Below the search bar, there are three search results:

- [PageRank en.wikipedia.org/wiki/PageRank](https://en.wikipedia.org/wiki/PageRank)
- See a page's importance using PageRank support.google.com
- PageRank Algorithm www.math.cornell.edu/

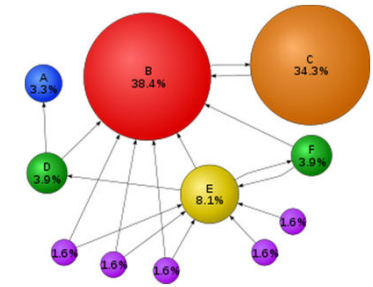
The BBC Four logo is visible in the top left corner of the video player interface. The video player also has navigation options for 'Home' and 'Clips'.

PageRank: algoritmus

- Google internetes keresőmotor legfontosabb eleme
- 1998: Larry Page, Sergey Brin
- Minden hiperlink felfogható egy-egy szavazatként a céloldalra
- Rekurzív definíció:
 - az a fontos oldal, amire fontos oldalak mutatnak



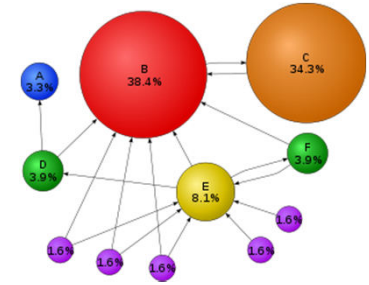
PageRank: definíció



$$\text{PageRank}(i) = (1 - d) + d \sum_{j \in M(i)} \frac{\text{PageRank}(j)}{L(j)}$$

- **M(i)**: azoknak az oldalaknak a halmaza, amik tartalmazznak linket az i. oldalra
- **L(j)**: a j. oldalról kimenő linkek száma
- **d**(csillapító tényezőt)
 - az oldalak a szavazatukból csak d részt osztanak tovább, (1-d)-t pedig megtartanak
- $\sum \text{PageRank}(i) = N$
 - A kezdetben a weblapok között egyenletesen elosztott fontosság átcsoportosítódik

PageRank: sztochasztikus szörföző

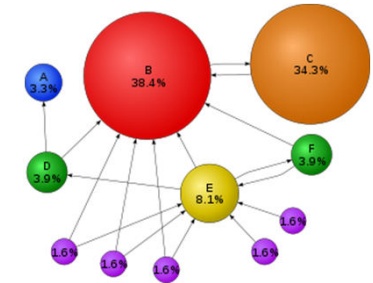
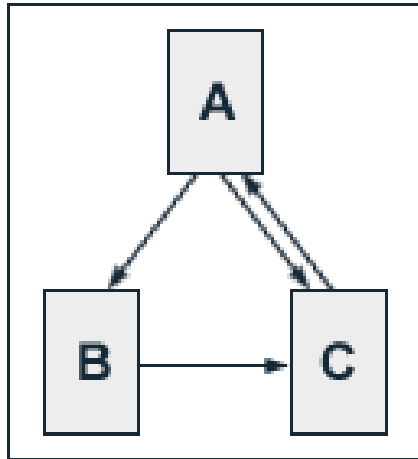


- PageRank: annak a valószínűsége, hogy odatalálunk az oldalra
 - véletlen bolyongás a hiperlinkek alkotta irányított gráfon

$$\text{PageRank}(i) = \frac{1-d}{N} + d \sum_{j \in M(i)} \frac{\text{PageRank}(j)}{L(j)}$$

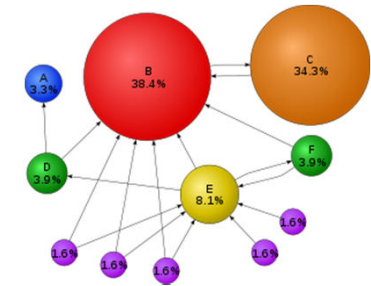
- Brin és Page:
 - eltévesztették a képletet, és az N nélküli változatot publikáltak
 - később kijavították, mégis a „hibás” változat terjedt el
 - N -t nehéz meghatározni

PageRank-algoritmus: PÉLDA



- $PR(A) = 0.5 + 0.5 PR(C)$
 $PR(B) = 0.5 + 0.5 (PR(A) / 2)$
 $PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))$
- $PR(A) = 14/13 = 1.07692308$
 $PR(B) = 10/13 = 0.76923077$
 $PR(C) = 15/13 = 1.15384615$
- $PR(A)+PR(B)+PR(C) = 3$

PageRank-algoritmus: EGYENLETRENDSZER



$$PR(A) = 0.5 + 0.5 PR(C)$$

$$PR(B) = 0.5 + 0.5 (PR(A) / 2)$$

$$PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))$$

$$\begin{cases} x_1 - 0.5 x_3 = 0.5 \\ -0.25 x_1 + x_2 = 0.5 \\ -0.25 x_1 - 0.5 x_2 + x_3 = 0.5 \end{cases}$$

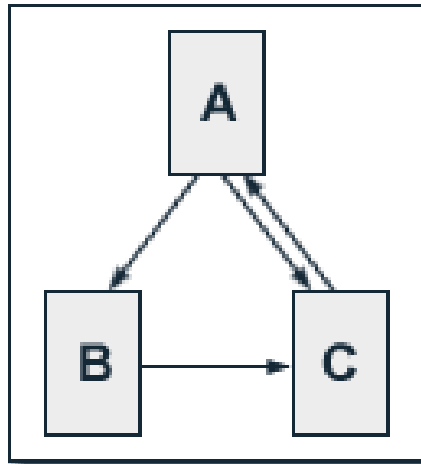
$A_0 (D_1)$

1	0	-0.5
-0.25	1	0
-0.25	-0.5	1

$x_1 = D_1 / D_0$
 $x_2 = D_2 / D_0$
 $x_3 = D_3 / D_0$

$A_1 (D_1)$	$A_2 (D_2)$	$A_3 (D_3)$
0.5 0 -0.5	1 0.5 -0.5	1 0 0.5
0.5 1 0	-0.25 0.5 0	-0.25 1 0.5
0.5 -0.5 1	-0.25 0.5 1	-0.25 -0.5 0.5

PageRank-algoritmus: EGYENLETRENDSZER



bemenet.txt

```
0.5
3
0 1 1
0 0 1
1 0 0
```

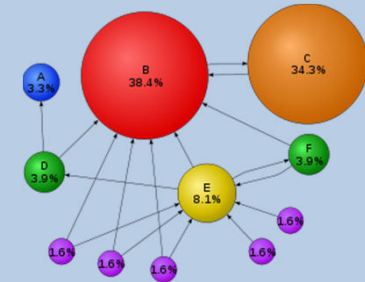
a

	0	1	1
	0	0	1
	1	0	1

A0

	1	0	-0.5
	-0.25	1	0
	-0.25	-0.5	1

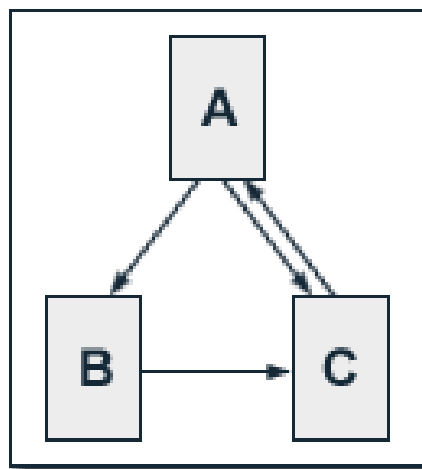
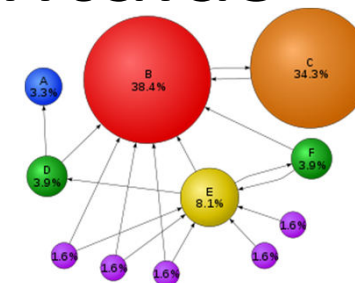
```
ifstream fin;
int main(){...
    int n,**a,i;
    double **A0,**Ai,*D,d;
    fin >> d >> n;
    D = letrehoz1Dtomb(n+1);
    a = letrehoz2Dtomb(n+1);
    A0 = letrehoz2Dtomb(n+1);
    Ai = letrehoz2Dtomb(n+1);
    beolvas(a,n);
    general_A0(a,n,d,A0);
    D[0] = determinans(A0,n);
    for ( i = 1 ; i <= n ; ++i ){
        general_Ai(A0,n,d,Ai);
        D[i] = determinans(Ai,n);
        cout << D[i] / D[0] << '\n';
    }...
}
```



**Determináns:
Gauss-elimináció**

PageRank: ITERATÍV implementálás

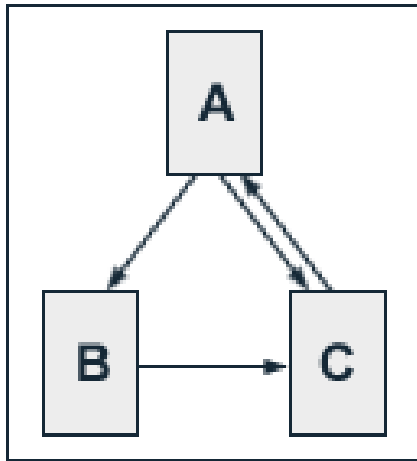
$$\text{PageRank}(i) = (1 - d) + d \sum_{j \in M(i)} \frac{\text{PageRank}(j)}{L(j)}$$



$$\begin{aligned} \text{PR}(A) &= 0.5 + 0.5 \text{PR}(C) \\ \text{PR}(B) &= 0.5 + 0.5 (\text{PR}(A) / 2) \\ \text{PR}(C) &= 0.5 + 0.5 (\text{PR}(A) / 2 + \text{PR}(B)) \end{aligned}$$

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.765625	1.1484375
3	1.07421875	0.76855469	1.15283203
4	1.07641602	0.76910400	1.15365601
5	1.07682800	0.76920700	1.15381050
6	1.07690525	0.76922631	1.15383947
7	1.07691973	0.76922993	1.15384490
8	1.07692245	0.76923061	1.15384592
9	1.07692296	0.76923074	1.15384611
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.15384615
12	1.07692308	0.76923077	1.15384615

PageRank: ITERATÍV implementálás



	bemenet.txt		
	0.5		
L	3		
2	0	1	1
1	0	0	1
1	1	0	0

```
#define ITnr 100
ifstream fin;
int main(){...
    int n,**a,nr,i;
    double *L,*PR, d;
    fin >> d >> n;
    PR = letrehoz1Dtomb(n); //PR[i]=1
    a = letrehoz2Dtomb(n); beolvas(a,n);
    L = letrehoz1Dtomb(n); kiszamol(a,n,L);
    for ( nr = 0 ; nr < ITnr ; ++nr ) {
        for ( i = 0 ; i < n ; ++i ) {
            frissit(i,PR,a,L,n,d);
        }
    }...
}
```

```
graph TD; B((B 38.4%)) <--> C((C 34.3%)); B --> A((A 3.3%)); B --> D((D 3.9%)); B --> E((E 8.1%)); B --> F((F 3.9%)); C --> E; D --> E; A --> E; F --> E; E --> P1((1.6%)); E --> P2((1.6%)); E --> P3((1.6%)); E --> P4((1.6%));
```

```
int s = 0;
for (j = 0 ; j < n ; ++j){
    if (a[j][i]) { s += PR[j]/L[j]; }
}
PR[i] = (1-d) + d * s;
```

Tanulmányozzuk a **PageRank**-algoritmus

- Mi a helyzet a lógó linkekkel (*dangling link*)?
 - zsákutca-oldalak, amelyek nem mutatnak sehova (ki-link számuk nulla)
- Hogyan befolyásolja a PR értékeket a d paraméter megválasztása?
 - Page és Brin: $d=0.85$
- <http://pr.efactory.de/e-pagerank-algorithm.shtml>