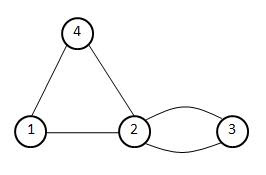
**Good Will Hunting Problem & Google PageRank mit Hilfe von Graphentheorie**

**Arbeitsphase 1:**

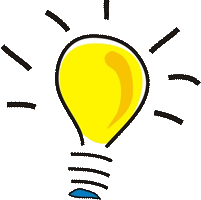


**Aufgabe 1:**

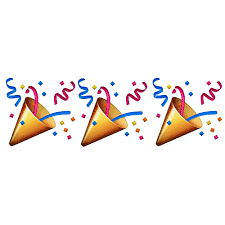
Konstruiert den abgebildeten Graphen in Maple und lasst euch die normale & die gewichtete Adjazenzmatrix ausgeben.

**Aufgabe 2:**

Berechnet die Matrix, die die Anzahl der Möglichkeiten angibt, wie man mit 3 Schritten von einer Ecke zu einer anderen Ecke kommt.

**Wichtige Befehle für Arbeitsphase 1:** 

|  |  |  |
| --- | --- | --- |
| Objekt | Maple-Befehl | Beschreibung |
|  |  |  |
| Ungerichteter Graph | G := Graph([1, 2, 3, 4], {[{1, 2}, 1], [{1, 3}, 2], [{2, 3}, 1]}) | Tupel G\_u = (E, V) bestehend aus einer (endlichen) Eckmenge E und einer (endlichen) Kantenmenge  V{{x, y}|x, y E; x y}.  Es gilt für die Kanten: {x,y}={y,x}. |
| Gerichteter Graph | H := Digraph([1, 2, 3, 4], {[[1, 2], 1], [[1, 3], 1], [[2, 3], 1]}) | Tupel G\_g = (E, V) bestehend aus einer (endlichen) Eckmenge E und einer (endlichen) Kantenmenge  V{{x, y}|x, y E; x y} von geordneten Tupeln, d.h {x,y}{y,x}. |
| Graph ausgeben | DrawGraph(G) |  |
| Adjazenzmatrix | AdjacencyMatrix(G) | Matrix, die speichert, welche Ecken des Graphen durch eine Kante verbunden sind.  🡪 Gibt an, wie viele Möglichkeiten es gibt von einer Ecke zu einer anderen Ecke zu kommen mit EINEM Schritt.  🡪 **Ai**(Matrixmultiplikation) gibt an, wie viele Möglichkeiten es gibt von einer Ecke zu einer anderen Ecke zu kommen mit i-Schritten. |
| Gewichtete Adjazenzmatrix | WeightMatrix(G) | Gibt nicht nur, DASS es eine Verbindung gibt, sondern auch wie stark gewichtet die Verbindung ist. |

CONGRATULATIONS! YOU’VE SOLVED THE FIRST 2 QUESTIONS OF THE GOOD WILL HUNTING PROBLEM!!

**Good Will Hunting Problem & Google PageRank mit Hilfe von Graphentheorie**

**Arbeitsphase 2:**

Angenommen das WWW hätte nur 4 Webseiten insgesamt (A, B, C, D) von denen wir den PageRank berechnen wollen.

Die Verlinkungsstruktur der Webseiten untereinander sieht folgendermaßen aus:

A🡪B

A🡪C

B🡪C

C🡪A

D🡪B

Erstellt den PageRank der Seiten A, B, C und D mit Hilfe der vereinfachten Version des Google-PageRank-Algorithmus. Was fällt euch auf?

Nutze hierzu das Matrix-Vektor-Iterationsverfahren!

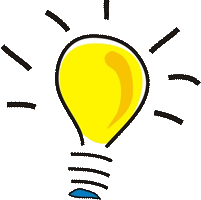
**Iterationsverfahren:**

**PR:** Rangvektor, der den PageRank aller Webseiten angibt,

d.h. Anzahl d. Einträge des Vektors Anzahl der Webseiten

n-mal

* **PRo** : PageRank im 0.Iterationsschritt: PR0 :=[ für n:=Anzahl Webseiten;
* **PRi** : PageRank im i-ten Iterationsschritt: PRi :=PRi-1\*A; für A=gewichtete Adjazenzmatrix

**Wichtige Befehle für Arbeitsphase 2:** 

|  |  |  |
| --- | --- | --- |
| Objekt | Maple-Befehl | Beschreibung |
|  |  |  |
| Allgemeine Iterationsfunktion | g:= x🡪x.A | Mit A: gewichtete Adjazenzmatrix |
| 1000. Iterationsschritt | (g@@1000)(P0) | @@: dient zur Hintereinanderausführung:  D.h. (g@@2)(P0 )=g(g(P0)=g(P1) |