You are allowed to work in groups of two. Only one submission per pair is necessary, but please indicate who you are working with. Submit your results as a zip-file to jpreusse@uni-koblenz.de until 16.07.2012, 11:59 pm.

1 Diffusion in Networks  (5 points)

1.1 Coordination Game

In this game every node has the choice between selecting either behavior ‘A’ or ‘B’. We represent this as injective function $f : V → \{ 'A', 'B' \}$. If two nodes $i$ and $j$ are linked with each other, they benefit from adopting their behavior to each other. This is modeled by a payoff $\text{payoff}_{ij}$ which is defined as follows:

$$\text{payoff}_{ij} = \begin{cases} 
  a, & \text{if } f(i) = f(j) = 'A' \\
  b, & \text{if } f(i) = f(j) = 'B' \\
  0, & \text{if } f(i) \neq f(j), 
\end{cases}$$

where $a, b > 0$ are the payoff values. All connected node pairs play this coordination game and a node’s overall payoff is defined as

$$\text{payoff}_i = \sum_{j \in N(i)} \text{payoff}_{ij},$$

where $N(i)$ is the set of nodes adjacent to $i$. Define $a_i$ and $b_i$ as the number of nodes in $N(i)$ that adopt behavior ‘A’ or ‘B’ respectively

$$a_i := \| \{ j | j \in N(i) \land f(j) = 'A' \} \|,$$

and

$$b_i := \| \{ j | j \in N(i) \land f(j) = 'B' \} \|.$$
and \( q := \frac{b}{a+b} \).

Node \( i \in V \) adopts behavior
\[
f(i) = \begin{cases} 
'A', & \text{if } i \in \text{initialAdopters} \lor \frac{a_i}{N(i)} \geq q \\
'B', & \text{if } \frac{a_i}{N(i)} < q 
\end{cases}
\]  

\[ (1) \]

### 1.1 Implementation

**Implement a function** with the signature

```matlab
function adoptedNodes = coordinationGame(adjacencyMatrix, initialAdopters, q)
```

which computes which nodes will adopt behavior 'A' after initial adopters changed to behavior 'A'. The method gets the adjacency matrix of a network and initial adopters as input. The parameter \( q \) is from Equation 1. Assume that all nodes in the original network select behavior 'B', whereas the initial adopters have switched to behavior 'A'. Starting from the initial adopters, you should now propagate the new behavior to their neighbors and their neighbors’ neighbors and so forth until no more nodes change their behavior.

For the network in Figure 1 the function call

```matlab
adoptedNodes = coordinationGame(mySparse('diffNet.txt', 0), [6,8,13], 1/2)
```

leads to the following diffusion of behavior in the network

```
adoptedNodes = 1 2 4 5 6 8 10 13
```

**Question 1** What if the initial adopters would change their behavior according to the same rules as the other nodes, so their behavior would not be fixed to 'B'? (1 point)

### 1.2 Modified Coordination Game

The rules of the modified version follow that of the coordination game, except that constants \( C_A \) and \( C_B \) are added to Equation 1. Now, node \( i \) adopts behavior \( Node \ i \in V \) adopts behavior
\[
f(i) = \begin{cases} 
'A', & \text{if } i \in \text{initialAdopters} \lor a_i \cdot a + C_a \geq b_i \cdot b + C_b \\
'B', & \text{if } a_i \cdot a + C_a < b_i \cdot b + C_b 
\end{cases}
\]  

\[ (2) \]

where \( C_A \) and \( C_B \) symbolize the individual benefits of using A and B, respectively.

### 1.2.1 Implementation

**Implement a function** with the signature

```matlab
function adoptedNodes = modifiedCoordinationGame(adjacencyMatrix, initialAdopters, a, b, C_a, C_b)
```

which computes which nodes will adopt behavior 'A' after initial adopters changed to behavior 'A'. The method gets the adjacency matrix of a network and initial adopters as input. The parameters \( a, b, C_a \) and \( C_b \) are from Equation 2. Assume that all nodes in the original network select behavior 'B', whereas the initial adopters have switched to behavior 'A'. Starting from the initial adopters, you
should now propagate the new behavior to their neighbors and their neighbors’ neighbors and so forth until no more nodes change their behavior.

For the network in Figure 1, the function call

```cpp
adoptedNodes = modifiedCoordinationGame(mySparse('diffNet.txt', 0), [7,8], 1, 2, 2, 1)
```

leads to the following diffusion of behavior in the network

```cpp
adoptedNodes = 2 3 7 8 9 12
```

**Question 2** Which effects do $C_a$ and $C_b$ have on the diffusion behavior in general?  (1 point)

2 Application  (5)

Imagine the following situation: there are exactly two companies $A$ and $B$ that each offer an operating system that is essential for every person in a network. For this task, we assume that every person uses exactly one operating system. Further, assume that the payoff for product $A$ is equal to product $B$ ($a = b$) and all users use $B$ at the beginning.

**Implement this and apply the diffusion mechanism** from the first task to the dataset ‘out.facebook-wosn-links’. For that, assume that 100 random users adopt $A$ unpaid (initial adopters).

**Question 3** How many users switch in average in 100 runs of the diffusion algorithm to $A$, as well?  (1 point)

2.1 Strategy  (2 points)

Suppose company $A$ is left with two options: double the payoff of its product ($a = 2 \cdot b$) or pay random people to use their product and thus diffuse it in the network. Both options come with a price: doubling the quality costs 100,000 € and paying a single person costs 100 €. For every user of their product company $A$ gets 10 € in return.

**Question 4** Having a budget of 100,000 €, which strategy is in average in 100 runs better for the Facebook dataset?  (1 point)

**Question 5** Having a budget of 200,000 €, which strategy is in average in 100 runs better for the Facebook dataset?  (1 point)

2.2 Choosing Initial Adopters  (2 points)

Given that $n = 200$ users are chosen as initial adopters, implement a strategy that is more efficient for choosing users than the random version for the Facebook dataset!