Applied Algorithm Design: Exam
June, 23rd, 2008

Prof. Pietro Michiardi

Exam Rules

- Exam Duration: 2 hours
- Documentation: none, except a 1-page cheat sheet
- **BOTH** exercises must be done
- Score is associated to every exercise and question. Sum is greater than 20

1 Exercise: gossiping

[POINTS = 8]

The following algorithm has been proposed during the last Lecture. This algorithm is used by every node i that belongs to a graph $G = (N, E)$, where $\{N\}$ is the set of nodes and $|N| = n$. We assume the graph G is connected, that is every node can communicate with any other node in the network. We further assume that there are no "routing" costs associated to message transmission.

- Initialization: $s_{0,i} = x_i$ and $w_{0,i} = 1$
- At iteration 0, a node i sends the couple $(s_{0,i} = x_i, w_{0,i} = 1)$ to itself

Algorithm 1: The Push-Sum protocol

```
Let $\{(\hat{s_r}, \hat{w_r})\}$ be all pairs sent to $i$ in round $t-1$
Let $s_{t,i} := \sum_r \hat{s_r}$, $w_{t,i} := \sum_r \hat{w_r}$
Choose a target $f_t(i)$ uniformly at random
Send the pair $(\frac{1}{2}s_{t,i}, \frac{1}{2}w_{t,i})$ to $f_t(i)$ and $i$ (yourself)
$\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average in step $t$
```

1. Explain which kind of algorithmic principle Algorithm 1 relies on.
[ANSWER]: Gossiping algorithms belong to the family of Randomized algorithms.

2. Explain how this algorithm can be modified to compute the sum of the readings at each node of the network $x_i \; \forall i \in N$
   [ANSWER]: There is no change to the algorithm. The initialization is different.

$$w_{0,i} = 0 \forall i \in \{N\} \setminus j$$

$$w_{0,j} = 1 \text{ for one node only}$$

3. Explain how this algorithm can be modified to compute the total number of nodes in the network
   [ANSWER]: Based on the previous question, you now know how to compute the sum. Instead of using the readings x_i from the nodes it is sufficient to set $x_i = 1$.

4. Explain how this algorithm can be modified to compute the maximum value of the readings, *i.e.* $\max_{i \in N} x_i$
   [ANSWER]: Assuming the same initialization as for Algorithm 1, see Algorithm **??** below:

Algorithm 2: The Push-MAX protocol

```
Let $\{(\hat{s_{r,j}}\}$ be all messages sent to $i$ in round $t-1$ from nodes $j$
Let $s_{t,i} := \max \hat{s_{r,j}}, s_{t-1,i}$
Choose a target $f_t(i)$ uniformly at random
Send the message $s_{t,i}$ to $f_t(i)$
$s_{t,i}$ is the estimate of the MAX in step $t$
```

5. For the the two questions above (2, 3, 4), provide a pseudo-code of the algorithm

2 Exercise: game theory

[POINTS = 8]

In the context of a 2-players **non** zero sum games, consider the following normal form game expressed in Table 1. The game can be defined as follows:

- Set of Players: player 1 and player 2
- Strategy set $S_i = \{A, B, C, D\}$
- Utility function, expressed in tabular form as follows

	A	B	C	D
A	3,2	2,0	0,2	5,5
B	2,1	-10,-1	0,0	-1,-1
C	1,5	1,2	0,2	1,3
D	8,3	4,5	2,4	0,-5

Table 1: Normal form representation of the game

1. Draw[1] at least two runs of the **Iterated Best Response** algorithm to find (if any) saddle point(s) of the game

    (a) How many saddle point are there in this game?

    (b) Which one is the most efficient? Which one is the worst?

    [ANSWER]: There are 2 saddle points $a = (4, 5)$ and $b = (5, 5)$, saddle point b is more efficient than a.

2. Sketch the pseudo-code of the iterated best response algorithm for this game

Based on the previous results, let us now consider a network creation game. The network creation game is defined as follows:

- The set of players is $\{N\}$, where $|N| = n$
- The strategy set available to player i is a binary vector: $[s_{i,1}, s_{i,2}, ...s_{i,n}]$, where $s_{i,i} = 0$ and where $s_{i,j} = 1$ iff player i establishes a link to player j
- The cost function to player i is the following:

$$c(i) = \alpha|s_i| + \sum_j d_G(i, j)$$

The solution to this game is a graph:

$$G(s) = (N, A(s)) \text{ where } A(s) = \{(i, j) : i \neq j, s_{ij} = 1 \vee s_{ji} = 1\}$$

1. Explain the influence of the parameter α
   [ANSWER]: If the parameter α is very small, it means that for a node buying a link is cost-less. Hence, to minimize the second term of the cost function, the best strategy is to lay down links to every other node. This implies that the equilibrium graph will be a clique. For higher α, the equilibrium graphs are stars and trees.

2. Sketch the pseudo-code for the iterated best response algorithm applied to this **multi-player** game. You can add some constraints to the problem, such as a bounded-degree.

3 Questions

1. `[2 points]` What is the Maximum Flow of a tree? Draw a tree example and explain.
   [ANSWER]: It is dictated by the bottleneck links from the root of the tree to the leaves.

[1] This means that you have to copy the matrix in Table 1 then you have to explicitly draw how cells of the table are traversed by the algorithm

2. [1 point] When does a bipartite graph have a perfect matching?
[**ANSWER**]: If $G = N, E$ and $N = X \cup Y$, where X and Y are disjoint sets of nodes connected by an edge with one end in X and the other end in Y, then you can have a perfect matching if $|X| = |Y|$

3. [4 points] Given a digraph $G = (V, E)$ and two nodes s and t, find the minimum number of edges whose removal disconnects t from s.

   (a) Given this problem, which algorithm can be used to find the solution?
   [**ANSWER**]: You can use a MAX-FLOW algorithm

   (b) Which is the running time of this algorithm?
   [**ANSWER**]: It depends, see execution times of Ford-Fulkerson or Edmonds algorithms

   (c) Why is this question is important in real systems?
   [**ANSWER**]: Because it tells you about the **robustness** of the graph G

4. [2 points] What is the importance of Flow Networks for modeling combinatorial problems? Use as a hint what you learned from the Reading material given in Lecture 4.
"Randomized decentralized broadcasting algorithms" Laurent Massoulie, Andy Twigg, Christos Gkantsidis, and Pablo Rodriguez, in Proc. of IEEE INFOCOM 2007
Another hint comes from considering the complexity and the nature of Matching Problems, and compare what we have done in Lecture 1 (Stable Matching, Bipartite Matching, Stable Marriages), to what we have seen in Lecture 4 (Maximum Flows, Minimum Cuts).
[**ANSWER**]: It is important beacuse it helps in transforming combinatorial problems in flow problems for which efficient algorithms exist.

5. [2 points] Define[2] the **0-1** Knapsack Problem.

   (a) (Optional) Define the **Fractional** Knapsack Problem.

   [**ANSWER**]: See Lecture notes

6. [1 point] What is Network Analysis? Can you give one example of centralities that you know and use everyday on the Internet?
[**ANSWER**]: It concerns the definition of graph centralities such as node, edge and group centralities as well as Network Statistics. A typical example is the degree distribution and the PageRank metric.

[2] When we say "define" it means that you have to provide a formal statement of the problem. It is required that you give a "text" definition such as "The knapsack problem is as follows: given a ..." and a mathematical definition. Recall that knapsack problems are related to maximization problems.