Seminars in Computer Networks
Homework 2

May 29, 2013

Full Name: _____ **Matricola**: _____

Instructions: You need to work on the homework individually.

Make sure that the solutions are typewritten or clear to read. *Give explanations for all of your claims.*

Hand in your solutions and keep a copy for yourself. After the due date we will post the solutions and in the final exam you may be asked to explain what were your mistakes.

Due date: 20/6/2013. On June 20, between 15:00 and 15:45, you can leave your solutions to the instructor in room B112 in via Ariosto (if unavailable, check also room B205). Alternatively, you can send them as a `.pdf` file by email; the address is on the website.

Problem 1. *Small world in the two-dimensional grid.* Consider a two-dimensional grid with n^2 nodes labeled with their coordinates: $(1,1)$, $(1,2)$, ..., $(n,n-1)$, (n,n). The distance between two nodes $u=(x_1,y_1)$ and $v=(x_2,y_2)$ is $d(u,v)=|x_1-x_2|+|y_1-y_2|$. Two nodes are considered adjacent in the grid when their distance is 1.

(a). What is the number of nodes of distance i from node $(1,1)$? Give a formula depending on i ($1 \le i \le n$).

(b). Let B_j be the set of nodes of distance at most 2^j from node $(1,1)$, excluding itself. Calculate the size of B_j for $0 \le j \le \log_2 n$.

(c). Let R_j be the set of nodes in B_j but not in B_{j-1}, that is, $R_j = B_j - B_{j-1}$ for $j \ge 1$, and $R_0 = B_0$. Compute a lower bound on the size of R_j. Specifically, if you obtain something of the form

$$a \cdot 2^s + b \cdot 2^t$$

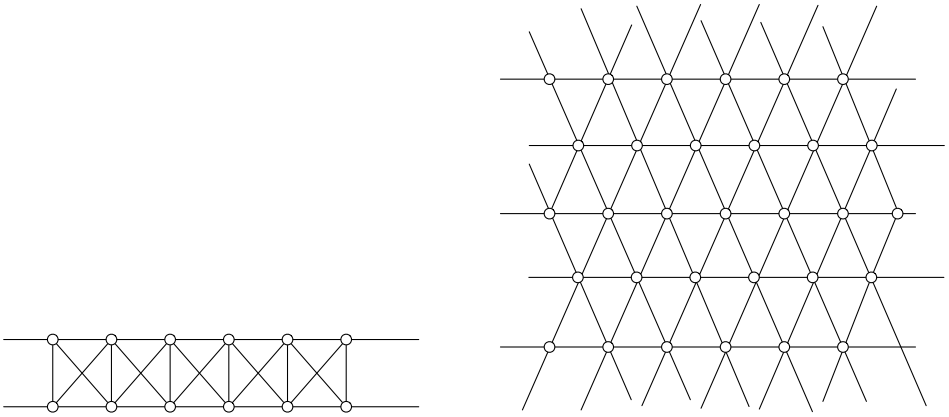
with $s > t > 0$ (and $1/2 < a, b \le 1$), you should report $a \cdot 2^s$ as your answer.

(d). Let $\alpha = 2$. According to the small-world model of Kleinberg, given that two nodes u and v are separated by distance r, the probability of the nodes being connected by a "long-distance" random link is lower bounded as follows:

$$\Pr[u \text{ has a long-distance connection to } v] \ge \frac{r^{-\alpha}}{4\ln(6n)}.$$

Given that we add to node $(1,1)$ only one long-distance link, use the result from (c) to lower bound the probability that node $(1,1)$ has a long-distance link to some node in R_j, for $0 \le j \le \log_2 n$. Does the answer depend on the value j?

Problem 2. *Cascades on infinite networks.* Find the cascading capacity of the two infinite networks drawn in Figure 1: (a) the infinite 5-regular "ladder", and (b) the infinite 6-regular triangular grid (note: a graph is *k-regular* if the degree of each node equals k).



(a) (b)

Figure 1: Two infinite networks

Problem 3. *Epidemics.* Imagine that you are advising a group of agricultural officials who are investigating measures to control the outbreak of an epidemic in its early stages within a livestock population. On short notice, they are able to try controlling the extent to which the animals come in contact with each other, and they are also able to introduce higher levels of sanitization to reduce the probability that one animal passes the disease to another.

Both of these measures cost money, and the estimates of the costs are as follows. If the officials spend x euros controlling the extent to which animals come into contact with each other, then they expect each animal to come into contact with

$$40 - \frac{x}{200\,000}$$

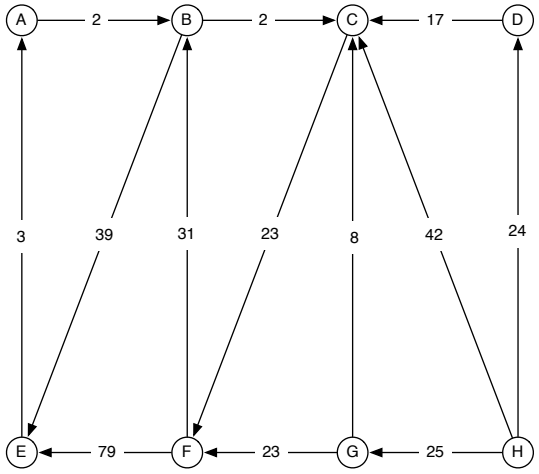
others. If the officials spend y euros introducing sanitization measures to reduce the probability of transmission, then they expect the probability an infected animal passes the disease to another animal contact to be

$$0.04 - \frac{y}{100\,000\,000}.$$

The officials have 2 million euros budgeted for this activity. Their current plan is to spend 1 million euros on each of the two kinds of measures. Using what you know about epidemics, would you advise them that this is a good use of the available money? If so, why? If not, can you suggest a better way to allocate the money?

Problem 4. *Bellman-Ford algorithm.* Consider the following edge-weighted digraph with 8 vertices and 13 edges.

$v \to w$	weight
A → B	2
B → C	2
B → E	39
C → F	23
D → C	17
E → A	3
F → B	31
F → E	79
G → C	8
G → F	23
H → D	24
H → G	25
H → C	42



Suppose that you run the Bellman-Ford algorithm to compute the shortest paths from H to every other vertex. Let `distTo[]` be the array that keeps track of the distance from H to the other nodes. Give the sequence of values in the `distTo[]` array immediately after the end of the first, then the second and then the third pass of the algorithm. Each pass consists of relaxing the 13 edges in the order given above. Here is the `distTo[]` array before the first pass.

v	A	B	C	D	E	F	G	H
`distTo[v]`	∞	∞	∞	∞	∞	∞	∞	0