UE Mathematics for Computer Science

First session exam December 18, 2013 (3 hours)
Only personal hand-written notes are allowed.
Use separated sheets for problems 1-2 (part I) and problems 3-4 (part II).
All problems are independent from each other.
Number of points given for each problem is given for information purposes only and is subject to modifications without notice.

Part I

Problem 1: Cover Time *(5 points)*

Consider a random walk on an undirected graph $\mathcal{G} = (X, E)$ of size $n = |X|$. For example, a flea is randomly jumping from node to node according to the jump probability

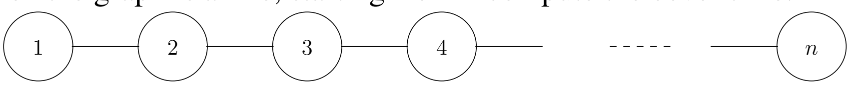
$$p_{i,j} = \frac{1}{d(i)},$$

where $d(i)$ is the degree of node i (number of neighbors).

The cover time of the random walk is the average time time needed by the flea to visit all the nodes of the graph

Question 1.1 : Line

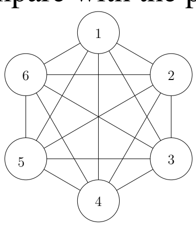
When the graph is a line, starting from 1 compute the cover time.



Hint: first show that the cover time is the hitting time of n starting from 1.

Question 1.2 : Complete graph

When the graph is a complete graph, starting from 1 compute the cover time and compare with the previous result.



Complete graph with $n = 6$ nodes

Hint: show that the problem is equivalent to the coupon collector problem.

Problem 2: Monotonicity *(5 points)*

Question 2.1 :

Compute $f(m,n)$ the number of functions from $\{1,2,\cdots,m\}$ to $\{1,2,\cdots,n\}$.

Question 2.2 :

Propose a simple algorithm that generates uniformly a function from $\{1,2,\cdots,m\}$ to $\{1,2,\cdots,n\}$.

Question 2.3 :

Compute the expected number of fixed points of a uniformly generated function from $\{1,2,\cdots,m\}$ to $\{1,2,\cdots,m\}$.

A function f is said to be **strictly increasing** if for all $x < y$ we have $f(x) < f(y)$.

Question 2.4 :

For $m \leqslant n$ use combinatorial arguments to compute $c(m,n)$ the number of strictly increasing functions from $\{1,2,\cdots,m\}$ to $\{1,2,\cdots,n\}$.

A function f is said to be **nondecreasing** if for all $x < y$ we have $f(x) \leqslant f(y)$.

Question 2.5 :

Use combinatorial arguments to compute $d(m,n)$ the number of nondecreasing functions from $\{1,2,\cdots,m\}$ to $\{1,2,\cdots,n\}$.

Question 2.6 :

Design an algorithm that generates uniformly a nondecreasing function from $\{1,2,\cdots,m\}$ to $\{1,2,\cdots,n\}$.

Question 2.7 : (bonus)

Compute the expected number of fixed points of a uniformly generated nondecreasing function from $\{1,2,\cdots,m\}$ to $\{1,2,\cdots,m\}$.

Part II

Problem 3: Fibonacci System Number *(6 points)*

Let us study the way the Fibonacci numbers can be used for representing integers. Let us write $j \gg k$ iff $j \geqslant k + 2$.

We will first prove the Zeckendorf's Theorem which states that every positive integer n has a unique representation of the form:

$n = F_{k1} + F_{k2} + ... + F_{kr}$ where $k1 \gg k2 \gg ...kr \gg kr$.

For instance, the representation of one million turns out to be:

$1000000 = 832040 + 121393 + 46368 + 144 + 55 = F_{30} + F_{26} + F_{24} + F_{12} + F_{10}$

Question 3.1 :

Show the existence by an induction on n. The proof is constructive using the following greedy rule: choosing F_{k1} as the largest Fibonacci number lower than n, then, choosing F_{k2} as the largest one that is less than $n - F_{k1}$ and so on...

Question 3.2 :

Show that this representation is unique.

Any unique system of representation is a number system. The previous theorem ensures that any non-negative integer can be written as a sequence of bits b_i, in other words,

$n = (b_m b_{m-1} ... b_2)_F$ iff $n = \Sigma_{k=2}^{m} b_k F_k$

Question 3.3 :

Write the Fibonacci representation of one million and compare it to the usual binary representation (recall that $1000000 = (2^{19} + 2^{18} + 2^{17} + 2^{16} + 2^{14} + 2^{9} + 2^{6})_2)$.

Conclude about their respective features. In particular, write the decomposition in the Fibonacci basis for the first 7 integers (starting from $1 = (0001)_F$). Give an argument for the property that there is no consecutive digits equal to 1 in such representations.

Question 3.4 :

Let us now study how to perform basic arithmetic operations within this system.

We will focus on the increment (addition of 1): obtaining $n + 1$ from n.

Detail first this operation when the last digit is 0 and justify it by the definition of Fibonacci numbers.

Give a process to obtain the increment when the two last digits are 01.

Problem 4: Miscellaneous Exercises *(4 points)*

This part contains two easy independent problems.

Question 4.1 :

Show by a geometrical proof that the odd square numbers are congruent to 1 modulo 8.

Question 4.2 :

$F(n)$ is the number of paths from node 1 to n in the following family of graphs of figure 1.

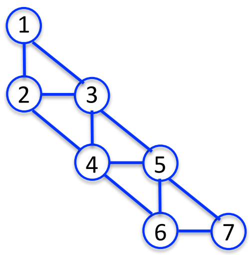


Figure 1: Counting paths from node 1 to node n ($n = 7$)

Show how this number is related to Fibonacci's numbers.