UE Mathematics for Computer Science

First session exam December 6, 2018 (3 hours)

Important information.
Read this before anything else!

- Only one personal hand-written sheet (2 pages) is allowed.
- Any printed document is not authorized during the exam, excepted dictionaries. Books are not allowed though.
- Please write your answers to each part on separate sheets of papers (2 separate sheets).
- The different exercises are completely independent. You are thus strongly encouraged to start by reading the whole exam. You may answer problems and questions in any order but they have to be written according to the original order on your papers.
- All exercises are independent and the total number of points for all problems exceeds 20. You can thus somehow choose the problems for which you have more interest or skills.
- The number of points alloted to each question gives you an estimation on the expected level of details and on the time you should spend answering.
- Question during the exam: if you think there is an error in a question or if something is unclear, you should write it on your paper and explain the choice you did to adapt.
- The quality of your writing and the clarity of your explanations will be taken into account in your final score. **The use of drawings to illustrate your ideas is strongly encouraged.**

Indicative grades

	Part I					Part II	
Exercises	1	2	3	4	5	1	2
points	2	2	2	4	3	6	6

Part I : Proofs and Recurrences

1. **Going further than triangular numbers**

   We established a strong link between the sum of odds $(2k+1)$ and the sum of two consecutive triangular numbers:

   $$(k+1)^2 = k^2 + 2k + 1, \text{ and } \sum_{k=0}^{n-1}(2k+1) = \Delta_n + \Delta_{n-1}.$$

   According to the binomial expression $(n+1)^3 = n^3 + 3n^2 + 3n + 1$, the next step is to compute the sum

   $$\sum_{k=0}^{n-1}(3k^2 + 3k + 1).$$

   How such numbers are linked with the sum of consecutive tetrahedral numbers ? [1]

2. **Graphical proof**

   Let $P(n) = n(n+1)(n+2)(n+3)$

   (a) Prove two following properties:

      1. $P(n) = k^2 - 1$ for some integer k.
      2. The squares of odd numbers are congruent to 1 modulo 8.

   (b) Deduce that $P(n)$ is a multiple of 8.

3. **Fibonacci numbers**

   Recall that Fibonacci's numbers are defined by the recurrence equation

   $$F(n+1) = F(n) + F(n-1), \text{ with the initial conditions } F(0) = F(1) = 1.$$

   (a) Prove the following property:

   $$\sum_{k=0}^{n-1} F(k) = F(n+1) - 1.$$

   (b) Deduce a closed form for computing $F(n_1) - F(n_2)$.

[1] We remind that tetrahedral numbers Θ_n are defined as the sum of triangular numbers.

4. **Master Theorem**

   Determining the cost of an algorithm for solving divide-and-conquer algorithm leads to a linear recurrence equation: Divide the input into a pieces of size $\frac{n}{b}$. The cost of the algorithm is given by the following equation:

   $$T(n) = a \cdot T(n/b) + f(n) \text{ with } T(1) = \Theta(1).$$

   Let assume that n is a perfect power of b and f is a polynomial $f(n) = n^\alpha$.

   Determine the closed expression of T in the following cases (the analysis should be detailed):

   1. $a = 2$, $b = 2$ and $\alpha = 0$.
   2. $a = 2$, $b = 2$ and $\alpha = 1$.
   3. $a = 2$, $b = 3$ and $\alpha = 1$.
   4. $a = 2$, $b = 2$ and $\alpha = 2$.

5. **Euclidian TSP**

   This exercise deals with the analysis of the approximation bound of the euclidian TSP problem (studied during the classes).

   Let remind the basic material: the problem is to determine a hamiltonian cycle of minimal weight (denoted by H^*) in an Euclidian weighted graph.

   The first step is to construct a spanning tree of minimum weight, say T^*.

   The second step is to construct a minimum perfect matching between the vertices of T^* that have an odd degree.

   Provide the proof of approximation of this step.

   More precisely, if C^* denotes such a minimum perfect matching, we ask you to prove that $\omega_{C^*} \leqslant \frac{1}{2}\omega_{H^*}$ where ω_G is the weight of the graph G (sum of the weights of its edges).

Part II : Counting and Coin Tossing

1. **Large Cuts in a Graph**

   Consider an undirected graph $\mathcal{G} = (X, E)$, $n = |X|$ is the number of nodes and $m = |E|$ is the number of edges. A cut $\mathcal{C} = \{A, \overline{A}\}$ of the graph is a partition of the set of nodes in two parts A and $\overline{A}$,

   $$A \cup \overline{A} = X \text{ and } A \cap \overline{A} = \emptyset$$

   (a) Compute the number of cuts of graph $\mathcal{G}$.

   An edge $e = (u, v)$ is said to connect A and $\overline{A}$ if either $u \in A$ and $v \in \overline{A}$ or $v \in A$ and $u \in \overline{A}$. The number $N(A, \overline{A})$ of such edges is called the size of the cut.

   (b) For a complete graph K_n (clique) compute the maximal and the minimal size of a cut.

   (c) For a line graph L_n compute the maximal and the minimal size of a cut.

   (d) When the size of A is n_A what is the maximum size of the cut.

   Suppose given a perfect coin (unbiased) denoted `coin()`

   (e) Design, prove and evaluate the complexity of a generator of a uniform cut in a graph $\mathcal{G}$.

   (f) Compute the expected size of a uniformly generated cut on a graph $\mathcal{G}$.

   (g) Deduce that there is at least one cut larger than $\frac{1}{2}m$ (half the number of edges).

   (h) Is this result coherent with the results found for the clique and the line.

2. **Hitting Times**

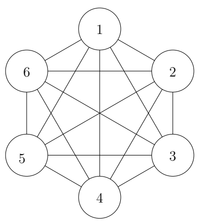
   Consider a random walk on an undirected graph $\mathcal{G} = (X, E)$ of size $n = |X|$. For example, a flea is randomly jumping from node to node according to the jump probability

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

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

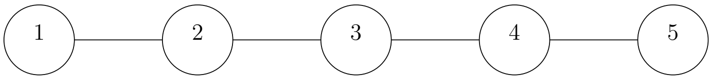
   The hitting time of node j by the random walk starting at node i, denoted by $T_{i \rightsquigarrow j}$, is the number of steps needed by the flea to reach j starting from i. We note the expected hitting time $t_{i \rightsquigarrow j}$.

   (a) When the graph is K_n, the complete graph (clique) with size n, compute the probability law of the hitting time $T_{i \rightsquigarrow j}$ and deduce its expectation.



Complete graph with $n = 6$ nodes

(b) When the graph is L_n, the line with n nodes, establish a recurrence equation on the expected times to hit n from i, $t_{i \rightsquigarrow n}$ and solve these equations for $n = 5$.



Line graph $n = 5$ nodes

(c) For an arbitrary graph with an adjacency matrix A, compute the coefficients of the transition probability matrix of the random walk. Then establish the system of n^2 equations verified by the $t_{i \rightsquigarrow j}$.

(d) Denote by $t^+_{i \rightsquigarrow j}$ the expected first time to return in i starting from i. For the line graph L_n establish a relation between $\mathbb{P}(t^+_{1 \rightsquigarrow 1} = 2k)$ and a Catalan's number.