Sélection internationale Session 2015
École Normale Supérieure Paris
Épreuve de culture scientifique - **Informatique** Durée : 3 heures

For candidates who chose **computer science** as **primary specialisation**

If you cannot answer a question, you may use it as hypothesis to later questions.
Calculators are not allowed.

Exercice 1. Suppose you are given an array $A[1 \dots n]$ of numbers, which may be positive, negative, or zero, and which are not necessarily integers.

1. Describe and analyze an algorithm that finds the largest sum of of elements in a contiguous subarray $A[i \dots j]$.

2. Describe and analyze an algorithm that finds the largest product of of elements in a contiguous subarray $A[i \dots j]$.

Exercice 2.

1. Describe and analyze an algorithm to sort an array $A[1 \dots n]$ by calling a subroutine SqrtSort(k), which sorts the subarray $A[k+1, \dots, k+\sqrt{n}]$ in place, given an arbitrary integer k between 0 and $n - \lceil \sqrt{n} \rceil$ as input. Your algorithm is only allowed to inspect or modify the input array by calling SqrtSort; in particular, your algorithm must not directly compare, move, or copy array elements. How many times does your algorithm call SqrtSort in the worst case?

2. Prove that your algorithm from question 1 is optimal up to constant factors. In other words, if $f(n)$ is the number of times your algorithm calls SqrtSort, prove that no algorithm can sort A using $o(f(n))$ calls to SqrtSort.

3. Now suppose SqrtSort is implemented recursively, by calling your sorting algorithm from question 1. For example, at the second level of recursion, the algorithm is sorting arrays roughly of size $n^{1/4}$. What is the worst-case running time of the resulting sorting algorithm?

   To simplify the analysis, you may assume that n, the array size, is of the form 2^{2^k}, so that repeated square roots are always integers.

Exercice 3.
For this question, we will need the following definitions.

Definition 1. A **graph** G is an ordered pair (V, E) where V is a set of vertices and E is a set of edges, 2-element subsets of V.

Definition 2. A **tree** is a connected graph with no cycles.

Definition 3. A graph is **planar** if it can be drawn in the plane in such a way that no two edges cross.

In a planar graph drawn in the plane, the plane is split into regions which we call *faces*. There is always exactly one "infinite" face which we call the *outer face*.

Definition 4. A **colouring** with k colours of a graph $G = (V, E)$ is assignment $c : V \to \{1, 2, \ldots, k\}$ such that adjacent vertices are assigned different values.

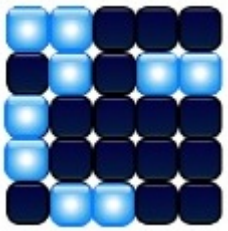
If such a c exists, we say that G is k-colourable.

1. Deduce the number of edges in a tree as a function of its number of vertices.
2. Show that if a planar graph has v vertices, e edges and f faces then $f + v = e + 2$.
3. Show the number of edges in a planar graph is bounded by a linear function of the number of vertices.
4. Show that every planar graph is 6-colourable.
5. Show, using Jordan theorem, that every planar graph is 5-colourable.

Exercice 4. A popular puzzle called "Lights Out!", made by Tiger Electronics, has the following description. The game consists of a 5×5 array of lighted buttons. By pushing any button, you toggle (on to off, off to on) that light and its four (or fewer) immediate vertical or horizontal neighbors.

The goal of the game is to have every light off at the same time. We generalize this puzzle to a graph problem. We are given an arbitrary graph G with a lighted button at every vertex. Pushing the button at a vertex toggles its light and the lights at all of its neighbors in the graph. A light configuration is just a description of which lights are on and which are off. We say that a light configuration is solvable if it is possible to get from that configuration to the all-off configuration by pushing buttons. Some (but clearly not all) light configurations are unsolvable.

1. Prove that the game reduces to the question whether we can find a subset X of the nodes, called an *activation set*, such that activating all nodes in X will turn off all nodes in G.
2. Give an *activation set* for the following configuration of the original "Lights out!" game:



3. Suppose the graph G is just a cycle of length n. Give a simple and complete characterization of the solvable light configurations in this case.
4. Characterize the set of solvable light configurations when the graph G is an arbitrary tree.

5. A grid graph is a graph whose vertices are a regular $h \times w$ grid of integer points, with edges between immediate vertical or horizontal neighbors. Characterize the set of solvable light configurations for an arbitrary grid graph. (For example, the original Lights Out puzzle can be modeled as a 5×5 grid graph.)

6. The goal of the next questions is to prove that the all-on configuration is solvable for any graph G (by induction on n, the number of nodes in G). Assume the claim is true for graphs with at most $n \geq 1$ nodes. Let G be a graph with $n+1$ nodes and for each node v in G let X_v be an activation set for the all-on configuration on the n-nodes graph $G \setminus \{v\}$. We assume that for each v in G, X_v is not an activation set for the all-on configuration on G.

   (a) Prove that if n is odd, there exist an activation set for the all-on configuration on G.

   (b) Show that if n is even, then at least one node v^* of G has even degree.

   (c) Show that if one pushes v^* and all nodes (with multiplicities) in all sets X_u for $u \in G$ which are not in the neighbourhood of v^*, one obtains the all-off configuration.

   (d) Conclude