Complexity

1 Solitaire Game

We consider the following game, named Solitaire Game. You are given an $m \times k$ board (m rows and k columns) where each one of the mk positions may be empty or occupied by either a red stone or a blue stone. Initially, some configuration of stones is placed on the board. This will be called the *initial configuration* of the Solitaire game. Now, for each column:

- if it contains both red and blue stones, then you **can** remove either all of the red stones in that column or all of the blue stones in that column;
- if it already contains only blue or only red stones, then you do not have to remove any further stones from that column.

The objective is to leave at least one stone in each row. If this objective can be obtained, we say that the game has a solution. In other words, the game has a solution if there is a sequence of decisions to be taken for every columns (between "remove red stones", "remove blue stones" and "remove nothing") which is such that in the end there is at least one stone per row. Finding a solution that achieves this objective may or may not be possible depending upon the initial configuration.

Question 1 *We consider the initial configuration in a 2 × 3 board (R and B respectively stand for Red and Blue stone, columns are numbered from 1 to 3, left to right):*

R	R	R
B	B	B

What if you decide to remove the red stone in every column? What if you decide to remove the red stone in two columns and a blue one in the remaining column?
What can we say if the initial configuration is:

	R	R
	B	B

Considering a 4 × 3 board with the initial configuration defined below. Is there a solution?

R		
B	B	R
B	R	
	B	B

We now define the corresponding decision problem:

SOLITAIRE
INSTANCE: an initial configuration of a Solitaire Game
QUESTION: does this game have a solution (in other worlds, is there a sequence of removign actions such that there is finally at least one stone in each row?)?

Question 2 *Prove that* **SOLITAIRE** *is in NP.*

We now proceed by a reduction form **3SAT**. For any instance ϕ of **3SAT** with m clauses $C_1, \ldots, C_m$ and k variables $x_1, \ldots, x_k$, we create an $m \times k$ Solitaire Game as follows:

- each column corresponds to a variable and each row to a clause;
- for every cell (i, j), we place a red stone in the cell if variable x_j occurs in C_i, and we place a blue stone in the cell if variable $\neg x_j$ occurs in clause C_i. Else, leave the cell empty.

Question 3 *What happens if one column of the initial configuration contains stones of the same color (all blue or all red) on every row? Explain why we can assume in the following that we do not consider this case anymore.*
Show that the construction is polynomial in m and k.

Question 4 *Create the instance from* **SOLITAIRE** *built from the following instance of* **3SAT**: $\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$. *Propose a truth assignment satisfying all clauses of ϕ.*
Write the instance ϕ' of **3SAT** *from which the following instance of* **SOLITAIRE** *was built.*

R	R	R	1
R	R	B	2
R	B	R	3
R	B	B	4
B	R	R	5
B	R	B	6
B	B	R	
B	B	B	

We now consider a positive instance of **3SAT** and the instance of **SOLITAIRE** which was created from it. If in a truth assignment which satisfies all clauses, x_j is set to True then remove all blue stones in the corresponding column, if x_j is set to False then remove all red stones in the column.

Question 5 *What can you say about the colors of the stone in one column? What is the boolean value of the* **literals** *(recalling that a red stone encodes an occurence of a literal x_j and a blue stone, an occurence of a literal $\neg x_j$) corresponding to the removed stones? Consider one arbitrary clause C_i and the correponding row i in the Solitaire game. Show that there is at least one stone on this row. Conclude.*

Conversely, we now assume that the instance of **SOLITAIRE** is positive. Recall that this means that all the stones in one column have the same color. We now define values for the boolean variable according to the following rule: Considering column j, if it has only red stones then wet set x_j to True, if it has only blue stones we set x_j to False.

Question 6 *Find a solution for the instance of* **SOLITAIRE** *built from ϕ in Question 4. Now set the boolean values to $x_1, \ldots, x_4$ according to the rule in the preceding question. What is the result for ϕ? Show that, in general, this rule defines a truth assignment which satisfy all clauses (note that a truth assignment must assign only one boolean value to every varaible). Conclude about* **SOLITAIRE**.

2 Hamiltonian circuit and path

The problem **HAMILTONIAN CIRCUIT** (or **HAMILTONIAN CYCLE**) is defined as follows:

HAMILTONIAN CIRCUIT (HC)
INSTANCE : a graph $G = (V, E)$ with $|V| = n$ (vertices are denoted v_i)
QUESTION : is there an hamiltonian circuit in G, that is, a permutation π of $\{1, \ldots, n\}$ s.t. $\forall i \in \{1, \ldots, n-1\}, (v_{\pi(i)}, v_{\pi(i+1)}) \in E$ and $(v_{\pi(n)}, v_{\pi(1)}) \in E$

The problem **HAMILTONIAN PATH** is defined as follows:

HAMILTONIAN PATH (HP)
INSTANCE : a graph $G = (V, E)$ with $|V| = n$ (vertices are denoted v_i), two vertices $s \in V, t \in V$
QUESTION : is there an hamiltonian path bewteen u and v in G, that is, a permutation π of $\{1, \ldots, n\}$ s.t. $\forall i \in \{1, \ldots, n-1\}, (v_{\pi(i)}, v_{\pi(i+1)}) \in E$ and $(v_{\pi(n)}, v_{\pi(1)}) \in E$ and $s = v_{\pi(1)}$ and $t = v_{\pi(n)}$ (the path starts in s and ends in t).

To prove that **HC** $\leq_P$ **HP**, the following polynomial reduction can be proposed. Given an instance $G = (V, E)$ of **HC**, instance $G' = (V', E')$ of **HP** is defined as follows:

- pick a vertex $v \in V$;
- $V' = V \cup \{v', s, t\}$;
- $E' = E \cup \{(v', x) | x \in V \text{ with } (v, x) \in E\} \cup \{(s, v), (t, v')\}$.

Question 7 *Draw a small example to illustrate the proposed reduction. Knowing that* **HAMILTONIAN CIRCUIT (HC)** *is NP-complete, show that* **HAMILTONIAN PATH (HP)** *is NP-complete.*