Exercise 1. Recall that a \textit{k-coloring} of a graph \(G = (V, E)\) is a function \(f : V \to \{1, 2, ..., k\}\) assigning colors to \(V\) such that no two adjacent vertices receive the same color.

\begin{tabular}{|l|}
\hline
\textbf{COLORING} \\
Input: & Graph \(G\), integer \(k\) \\
Question: & Does \(G\) have a \(k\)-coloring? \\
\hline
\end{tabular}

\begin{itemize}
\item Suppose \(A\) is an algorithm solving \textsc{Coloring} in \(O(f(n))\) time, \(n = |V|\), where \(f\) is non-decreasing. Design a \(O^*(f(n))\) time algorithm \(B\), which, for an input graph \(G\), finds a coloring of \(G\) with a smallest number of colors.
\end{itemize}

Exercise 2. Recall that a graph \(G = (V, E)\) is \textit{bipartite} if \(G\) has a 2-coloring. A \textit{matching} in a graph \(G = (V, E)\) is a set of edges \(M \subseteq E\) such that no two edges of \(M\) have an end-point in common. The matching \(M\) in \(G\) is \textit{perfect} if every vertex of \(G\) is contained in an edge of \(M\).

\begin{tabular}{|l|}
\hline
\textbf{\#Bipartite Perfect Matchings} \\
Input: & Bipartite graph \(G = (V, E)\) \\
Output: & The number of perfect matchings in \(G\) \\
\hline
\end{tabular}

1. Design an algorithm for \#\textsc{Bipartite Perfect Matchings} with running time \(O^*\left(\left(\frac{n}{2}\right)!\right)\), where \(n = |V|\).

2. Design a polynomial-space \(O^*(2^n/2)\)-time inclusion-exclusion algorithm for \#\textsc{Bipartite Perfect Matchings}.