- Primal/Dual Algorithms. -

- Exercice 1 -
In ancient times, before LP-solvers, people from Budapest used to solve their assignment problems by the so-called "Hungarian Method". This is the first primal/dual algorithm designed by mankind. Assume that $M = (m_{ij})$ is an $n \times n$ matrix with non negative integer values, the goal is to find a permutation $\sigma$ such that the sum of all $m_{i\sigma(i)}$ is minimized. The algorithms consists in iterating the following :

1. While there exists a row $R$ (resp. a column $C$) whose minimum entry $m$ is non zero, subtract $m$ from all entries of $R$ (resp. of $C$).
2. If there exists a permutation $\sigma$ such that $m_{i\sigma(i)} = 0$ for all $i = 1 \ldots n$, return $\sigma$.
3. Else find a set $S$ with less than $n$ rows and columns covering all 0 entries. Find the smallest entry $m$ not covered by $S$, subtract $m$ from every row not in $S$, and add $m$ to every column in $S$.

1. Apply this algorithm to the following 5 by 5 matrix (starting to deal with the rows in the while) :

\[
\begin{pmatrix}
2 & 3 & 4 & 6 & 8 \\
5 & 5 & 7 & 2 & 3 \\
6 & 3 & 1 & 2 & 2 \\
7 & 5 & 4 & 3 & 6 \\
8 & 7 & 5 & 3 & 2
\end{pmatrix}
\]

2. Is the first "While" of the algorithm useful ?
3. Show that the set $S$ in the "Else" always exist.
4. Show that the algorithm terminates, i.e. always return a permutation $\sigma$.
5. Show the minimality of $\sigma$ by providing a dual certificate.
6. Interpret the Hungarian Method as a primal/dual algorithm, where we iterate improvements on a dual certificate $y$ (initially set to 0).

- Exercice 2 -
We have seen that Dijkstra's Algorithm is indeed a primal/dual algorithm when the input graph is undirected. What about solving the minimal length directed $st$-path when the input is a directed graph ?

- Exercice 3 -
Use the primal/dual algorithm to find a minimum cost arborescence rooted at vertex 1 for the weighted directed graph given by the following adjacency matrix (here . means no arc) :

\[
\begin{pmatrix}
. & 3 & 4 & 5 \\
. & . & 2 & 1 \\
. & 2 & . & 3 \\
. & 1 & 3 & .
\end{pmatrix}
\]

- Exercice 4 -
In the Steiner Tree Problem (STP), we are given a graph $G = (V, E)$, a set $X$ of vertices, and a non negative weight function $c$ on $E$. The goal is to find a minimum cost subtree of $G$ containing $X$.

1. Show that STP is NP-hard (reduction from set-cover).
2. Provide a violation (multi)oracle $\nu$ for STP which gives a primal/dual algorithm achieving a 2-approximation.