Solution.

Exercise 1.

1. If we interpret each x_e as 1 if the edge belongs to a matching and 0 otherwise, a feasible solution to this linear program with the additional constraint $x_e \in \{0,1\}$ is indeed a matching. This comes from the fact that the condition implies that we cannot take two edges with a common endpoint.

   The matrix of the linear program contain as many rows as vertices and as many columns as edges. It is the vertex-edge incidence matrix of the graph. Since an edge connects only two vertices, each column contains exactly 2 ones. Moreover, for a bipartite graph, we can split the lines in two parts such that each columns contain exactly 1 one in each part.

   Now, for any submatrix :

   - if it contains an all 0 column, its determinant is zero.
   - if all columns have two ones, we can take the sum of all the rows in the first part minus all the row in the second part. We obtain a non-zero linear combinaison that is equal to 0 and the determinant is still zero.
   - if none of the above cases hold, then there is a column with only one 1. We can develop the determinant according to this column and obtain plus or minus the determinant of a smaller square submatrix. We finish the proof by induction on the size of the submatrix.

   Since the matrix is totally unimodular, all the extremal points in the convex set of feasible solutions have integer coordinates. Remark that we need the weight to be an integer to conclude this. Hence an optimal solution exists with integer coordinates and can be found by solving the linear program.

2. The dual of this program is :

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_v y_v \\ \text{subject to} \quad & \textstyle\sum_{v, v \in e} y_v \geq w_e \quad \forall e \in E \\ & y_v \geq 0 \quad \forall v \in V. \end{aligned}$$

   The matrix being the transpose of the one in (1), it is still totally unimodular. If the weights are 1, a feasible solution to this program corresponds to a vertex cover since the condition implies that each edges is adjacent to at least one vertex with $y_v = 1$.

3. Using the duality theorem of LP, we can infer than on a bipartite graph, the cardinality of a maximum matching is the same as the cardinality of a minimum vertex-cover.

Exercise 2. We will consider the following greedy approximation algorithm

1. Start with an empty set of edges E' and all x_e and y_v equal to 0.
2. While edges with no common endpoints with edges in E' exist, add to E' the one $e = (u, v)$ of maximum weight. Set $x_e = 1$ and set $y_u = y_v = w_e$.
3. return E'.

Remark first that the algorithm returns a feasible matching. Moreover, at the end of the algorithm, if y_v is non-zero, it means we added an edge adjacent to v and so the primal complementary slackness condition is satisfied. That is $y_v \neq 0 \Rightarrow \sum_{e, v \in e} x_e = 1$. For each edge e we can show that $\sum_{v, v \in e} y_v \geq w_e$ since if we have $x_e = 1$ we know $\sum_{v, v \in e} y_v = 2w_e$ and if $x_e = 0$ then $e = (u, v)$ was not selected which implies that an edge adjacent to u or v was selected with weight greater than w_e.

In conclusion, the relaxed complementary slackness condition is satisfied with a factor of 2, and the algorithm is a factor 2-approximation to the maximum weight matching in a general graph.

Exercise 3. Since Dijkstra's algorithm computes the shortest path from s to every destination, we will use the following primal program:

Let $G = (V, E)$ be a directed graph, and $c : E \to \mathcal{R}^+$ a cost function on the edges (their weight). We will use one variable x_e per edges. The problem can be formulated as

$$\begin{array}{rl} \text{minimize} & \sum_e x_e c(e) \\ \text{subject to} & \forall v \in V \quad \sum_{e=(u,v)} x_e - \sum_{e=(v,u)} x_e \geq b_v \\ & \forall e \in E \quad x_e \geq 0 \end{array}$$

where the b_v are all 1 except $b_s = -n + 1$. Basically, we want to send a flow of 1 from s to every node. Hence each node consumes one unit of flow, s produce $n - 1$ units and x_e represents the value of the flow on the corresponding edges. If we fix one shortest path per node, it is easy to see that a solution where x_e is the number of such paths that use e is a feasible solution.

The dual program of this program is given by

$$\begin{array}{rl} \text{maximize} & \sum_{v \neq s} y_v - (n-1) y_s \\ \text{subject to} & \forall e = (u, v) \in E \quad y_v - y_u \leq c(e) \\ & \forall v \in V \quad y_v \geq 0 \end{array}$$

where y_u has to be interpreted as the distance between s and u. We will in particular assume without loss of generality that $y_s = 0$.

We can then see Dijkstra's algorithm as a primal-dual algorithm. Suppose we start with all x_e equal to 0 and all y_v equal to ∞.

At each step, Dijkstra's algorithm chooses a new node v to settle. After this operation, y_v is set to the maximum value such that all the condition of the primal involving v are satisfied. In particular, one of the conditions gets tight. Remark that Dijkstra's algorithm choose this node v as the one leading to the smallest value, hence this value will always satisfy the constraint in the future. This is because all the node settled after it will have a higher distance to s.

Moreover, we can increase by one all the x_e on a shortest path from s to this settled node. The condition associated to v becomes true, and the previous satisfied conditions of the primal program are still satisfied.