**Exercise 10.3.** After your glorious app PikPok hit number 1 in the app store, you're preparing for version 2. Obviously, it needs to be great.

You've gathered a list of $k$ features $F_1, \ldots, F_k$ that you could potentially add to version 2. However, there are complicated dependencies and requirements among them so you don't necessarily want to add all of them. There are 3 types of specifications defined over pairs of features $F_i$ and $F_j$:

1. Requirements: Your app must include either $F_i$ or $F_j$.

2. Conflicts: You cannot include both $F_i$ and $F_j$.

3. Dependencies: If you include $F_i$, then you must include $F_j$.

Collectively, we call requirements, conflicts, and dependencies the *feature specifications*. The feature specifications are given in list form. The high-level task is to decide which of the features to implement, based on the given feature specifications. We have two versions of the problem:

1. In the idealistic feature selection problem, the task is to decide if there is a subset of features that satisfies all the feature specifications.

2. In the realistic feature selection problem, the task is to choose a subset of features that satisfies the maximum number of feature specifications.

For each of the problems above, either (a) design and analyze a polynomial time algorithm (the faster the better), or (b) prove that a polynomial time algorithm would imply a polynomial time algorithm for SAT.

---

**Exercise 10.3.1.** Exercise

---

*Solution.* solution        □

**Exercise 10.3.2.** Exercise

*Solution.* solution      □