Exercise 2

Due: 19/2/2024

1. Given a set $A \subseteq \{1, ..., m\}$ of size $|A| = n$, and an integer T, find an $O(nm \log^2 m)$ time algorithm that computes whether there exists a subset $S \subseteq A$ such that $\sum_{a \in S} a = T$. (*Hint: define a polynomial for every element in A.*)

2. Let $G = (V, E)$ be graph with n vertices. Suppose you are given an oracle that can answer whether a graph has a perfect matching. Show how to find the size of a maximum matching in G using only $O(\log n)$ calls to the oracle. You may construct a graph with $O(n)$ vertices and $O(n^2)$ edges for each oracle call. (*Hint: use binary search.*)

3. Let $G = (V, E)$ be a graph, and let $M \subseteq E$ be a matching such that there is no augmenting path of length at most 3 for M. Prove that $|M| \geq \frac{2}{3}|M^*|$, where M^* is the maximum matching for G.

4. Let $G = (V, E)$ be a connected graph with n vertices and m edges, and with weights $w : E \to \mathbb{R}$ on the edges. The *widest path* P between two vertices $u, v \in V$, is a path that maximizes $\min_{e \in P}\{w(e)\}$. Show that there is a collection of widest paths for all $\binom{n}{2}$ pairs of vertices which is a tree, and that this tree can be computed in $O(m \log \log n)$ time.