Exercise 11.8 (Approximating subset sum.) Let $\epsilon \in (0,1)$ be fixed. Here we treat ϵ as a fixed constant (like $\epsilon = .1$, for 10% error); in particular, running times of the form $O(n^{O(1/\epsilon)})$ count as a polynomial.

A $(1 \pm \epsilon)$-approximation algorithm for subset sum is one that (correctly) either:

1. Returns a subset whose sum lies in the range $[(1-\epsilon)T, (1+\epsilon)T]$.
2. Declares that there is no subset that sums to (exactly) T.

Note that such an algorithm does not solve the (exact) subset sum problem.

Note. You may (and should) assume $T, x_i \in \mathbb{R}_{\geq 0}$. The problem appears (computationally) hard otherwise.

Exercise 11.8.1. Suppose every input number x_i was "small", in the sense that $x_i \leq \epsilon T$. Give a polynomial time $(1 \pm \epsilon)$-approximation algorithm for this setting.

Solution. □

Exercise 11.8.2. Suppose every input number x_i was "big", in the sense that $x_i > \epsilon T$. Give a polynomial time $(1 \pm \epsilon)$-approximation algorithm for this setting.

Solution. $\square$

Exercise 11.8.3. Now give a polynomial time $(1 \pm \epsilon)$-approximation algorithm for subset sum in the general setting (with both big and small inputs).

Solution. □