Imagine that you have a robot arm consisting of a sequence of rigid line segments $L_1, L_2, \ldots, L_n$ hinged together consecutively. The lengths of the line segments $l_1, l_2, \ldots, l_n$ are positive integers. You may rotate freely around the hinges and the line segments are allowed to cross over. Our goal is to pack in the line segments into one line segment as compactly as possible. More precisely, given positive integers $l_1, l_2, \ldots, l_n$ (and $l_0 = 0$) and a sequence of $+1$'s, $s_0, s_1, \ldots, s_n$, where $s_i = 1$ or $s_i = -1$, $0 \leq i \leq n$, define

$$m_1 = \min_j \left( \sum_{i=0}^j s_i l_i \right), \quad m_2 = \max_j \left( \sum_{i=0}^j s_i l_i \right)$$

(where $0 \leq j \leq n$).

Find $\min_s (m_2 - m_1)$, where the minimum is taken over all sequences $s = (s_0, s_2, \ldots, s_n)$.

1. Describe an algorithm that solves this problem.

2. Write and test a program to implement your algorithm. Provide enough test cases to convince us that your program works correctly.

3. Analyze your algorithm.

4. Are you happy with the performance of your algorithm? If yes, explain. If not, what do you think is the underlying reason why your algorithm is not efficient, namely what is the complexity of the problem? Can you make an assumption on the input that allows you to come up with a more efficient algorithm?