The given algorithm is NOT guaranteed to minimize badness.

A counterexample is that, suppose there are three "clients." The three bodies have 1, 15, 32 feet, respectively. The three caskets are 16, 31, 50 feet, respectively. The given algorithm will match bodies to caskets as follows:

body	casket
15	16
32	31
1	50

Thus the total badness of fit is $|15-16|+|32-31|+|1-50| = 1+1+49 = \mathbf{51}$. However, if we match them as shown below:

body	casket
1	16
15	31
32	50

This will reduce the total badness to $|1-16|+|15-31|+|32=50| = 15+16+18 = \mathbf{49}$, which is smaller than 51. Because there is at least one way that gives smaller total badness of fit, we can conclude that the given algorithm is not optimal.

My algorithm works like this: sort bodies by length so that $l_1 \leq l_2 \leq ... \leq l_n$; also sort caskets by length so that $m_1 \leq m_2 \leq ... \leq m_n$. Then, assign l_1 to m_1, l_2 to m_2, and so on, until l_n is assigned to m_2.

- **Greedy Choice**: Let the shortest body to have length l_1 and the shortest casket m_1. Then there exists an optimal solution to P that matches l_1 to m_1.

  Pf. Suppose a solution does not match l_1 to m_1. Then it must match l_1 to some other casket and match m_1 to some other body. Without loss of generality, we can match l_1 to l_2 and match m_1 to m_2.

  - badness of fit with l_1 to m_1 and l_2 to m_2: A= |l$_1$ - m_2 | + |l$_2$ - m_1 |.
  - badness of fit with l_1 to m_2 and l_2 to m_1: B= |l$_1$ - m_1 | + |l$_2$ - m_2 |.

  There are four cases in total, two of them having two sub-cases. Both A and B can be written in respect to l_1, l_2, m_1 and m_2 without the absolute value sign.

  - **Case 1**: $l_1 \geq m_1$, $l_2 \geq m_2$.
    - When $m_2 \geq l_1$. So, $m_1 \leq l_1 \leq m_2 \leq l_2$
      A=$m_2 - l_1 + l_2 - m_1$, B=$l_1 - m_1 + m_2 - l_2$
      $A - B = 2(m_2 - l_1) \geq 0$, so $\mathbf{A \geq B}$
    - When $m_2 \leq l_1$.
      A=$l_1 - m_2 + l_2 - m_1$, B= $l_1 - m_2 + l_2 - m_1$
      So $\mathbf{A = B}$
  - **Case 2**: $l_1 \leq m_1$, $l_2 \leq m_2$.
    - When $m_1 \leq l_2$. So, $l_1 \leq m_1 \leq l_2 \leq m_2$
      A=$m_2 - l_1 + l_2 - m_1$, B=$m_1 - l_1 + m_2 - l_2$
      $A - B = 2(l_2 - m_1) \geq 0$, so $\mathbf{A \geq B}$
    - When $m_1 \geq l_2$. So, $l_2 \leq l_1 \leq m_2 \leq m_1$
      A=$m_2 - l_1 - l_2 + m_1$, B=$m_2 - l_1 - l_2 + m_1$
      So $\mathbf{A = B}$.

- **Case 3** $l_1 \geq m_1$, $l_2 \leq m_2$. So, $m_1 \leq l_1 \leq l_2 \leq m_2$
  A=$m_2 - l_1 + l_2 - m_1$, B=$l_1 - m_1 + m_2 - l_2$
  $A - B = 2(l_2 - l_1) \geq 0$, so $\mathbf{A} \geq \mathbf{B}$
- **Case 4**: $l_1 \leq m_1$, $l_2 \geq m_2$. So, $l_1 \leq m_1 \leq m_2 \leq l_2$
  A=$m_2 - l_1 + l_2 - m_1$, B=$m_1 - l_1 + l_2 - m_2$
  $A - B = 2(m_2 - m_1) \geq 0$, so $\mathbf{A} \geq \mathbf{B}$

To sum up, each of the six cases gives either $A = B$ or $A \geq B$, which means that there is no other more optimal solution that matches m_1 to l_1. Therefore, the optimal solution must contain this pair.

- **Inductive Structure**: After matching a pair of body and casket, we are left with a subproblem P' without external constraints.

  Pf. After matching k pairs of bodies, we have (n-k) bodies and (n-k) bodies in sorted order that can be matched one-to-one. There is no external constraint on how to match them.

- **Optimal Substructure**: Badness of fit for problem P is simply the sum of badness of fit for subproblem P' and that of the $(k+1)$th greedy choice. Applying the standard contradiction argument, we conclude that if the sum is minimum, then after adding badness of fit for the last greedy choice the final solution will also be optimal.

Complexity Analysis The sorting process requires O(n log n) time. Matching between bodies and caskets takes O(n) time by linearly iterating through two arrays. The overall time complexity of the algorithm is **O(n log n)**.