Let's label our solution, from the greedy algorithm: **G: {**(a1,b1) , (a2,b2)…(a_n,b_n)**}** where n is the total number of pairs.

Let's label our solution, from the optimal algorithm: **O: {**(a1,b1) , (a2,b2)…(a_n,b_n)**}** where n is the total number of pairs.

In both the cases, all the elements will exist in both sets as the both sticks are of the same number, *n*.

However, the ordering can be different in both the sets.

Let's take an example:

G: {(2,5) (5, 10)} => cost = (3+5)/2 => 4

O: {(5,5) (2, 10)} => cost = (0+8)/2 => 4

As we can see, both algorithms generate same cost hence the greedy algorithm does as good as the optimal one.

If we exchange pairs between G and O, we can continue doing so without worsening the quality of the algorithm, hence the greedy solution is the optimal one.