Fall 2014: You are given n rods; they are of length $l_1, l_2, \ldots, l_n l_1, l_2, \ldots, l_n$, respectively. Our goal is to connect all the rods and form a single rod. The length after connecting two rods and the cost of connecting them are both equal to the sum of their lengths. Give an algorithm to minimize the cost of connecting them to form a single rod. State the complexity of your algorithm and prove that your algorithm is optimal.

The solution is to keep connecting the smallest two rods or sets of already connected rods. Implementation: Place all rods in a min heap with their key values representing their length. At each step extract two elements from the set and insert the combined rod back into the min heap.
This takes a total of O(n log n) time

Fact 1: the cost contribution of a rod i to the total assembly cost is Length(i) * Level(i), where Level(i) is the level at which the rod is first assembled with other rods (level 1=root level, level 2= level below root, etc.).
Proof: By observation of cost of assembly tree

Fact2: There is an optimal assembly tree of rods in which the two smallest rods are leaf nodes at the lowest level of the tree and the children of the same parent.
Proof: let's say the two smallest rods are not leaf nodes at the lowest level of the tree in the optimal solution, using fact 1, we can swap these rods with two rods at the lowest level of the tree and thereby reducing the total cost of the assembly tree.
Also, if the two rods are leaf nodes at the lowest level but not children of the same parent we can swap two rods to make these rods children of the same parent without changing the total cost of the tree (again based on Fact 1).

Assume that there is an optimal assembly tree T* and our solution produces tree T. We will show that our tree T is also optimal.
To do this, we apply fact 2 to T* and move the smallest rods to the lowest level of the tree as children of the same parent—without increasing the cost of T*.
We then eliminate the two smallest rods at the bottom of the two trees(since these are the first two rods that are assembled in our algorithm) and assume that there is a new combined rod in place of their parent node. We will get two new trees T*' and T' which have the same set of bars at the leaf nodes (in other words, we can think of these as solutions to a new problem with n-1 bars where the two smallest bars in the previous problem are now combined into the same bar in the new problem).

We now repeat the above step (applying fact 2 and eliminating the two smallest rods from the optimal assembly tree and our tree) recursively. We will find an optimal solution that follows the same exact assembly sequence that is found in our algorithm. Therefore we can state that our algorithm also produces an optimal assembly tree.