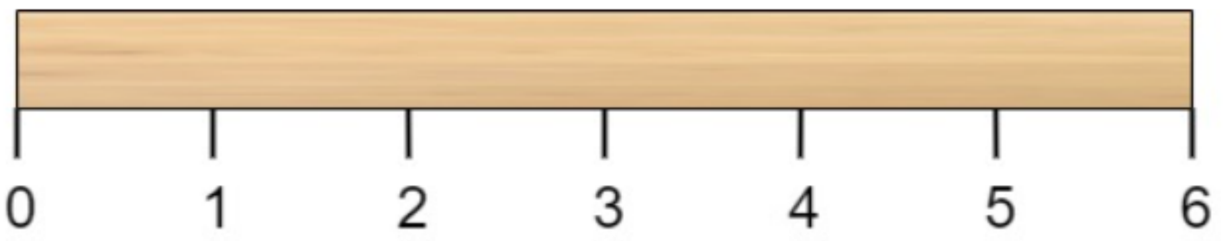
Lab Sheet 5

Problem 1:
Given a wooden stick of length n units. The stick is labelled from 0 to n. For example, a stick of length **6** is labelled as follows:



Given an integer array cuts where cuts[i] denotes a position you should perform a cut at.

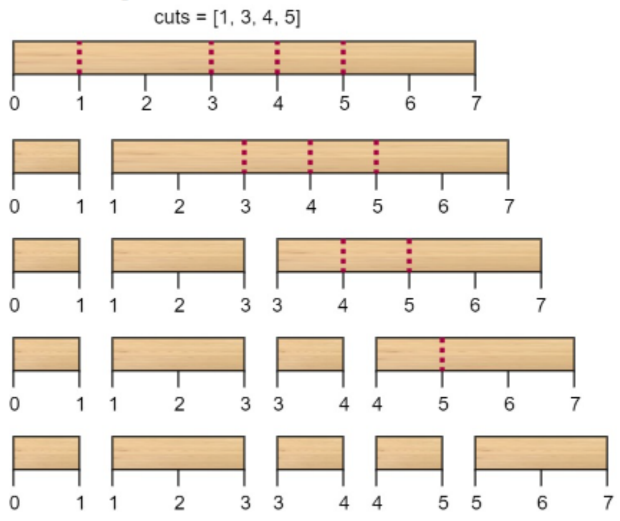
You should perform the cuts in order, you can change the order of the cuts as you wish.

The cost of one cut is the length of the stick to be cut, the total cost is the sum of costs of all cuts. When you cut a stick, it will be split into two smaller sticks (i.e. the sum of their lengths is the length of the stick before the cut). Please refer to the first example for a better explanation.

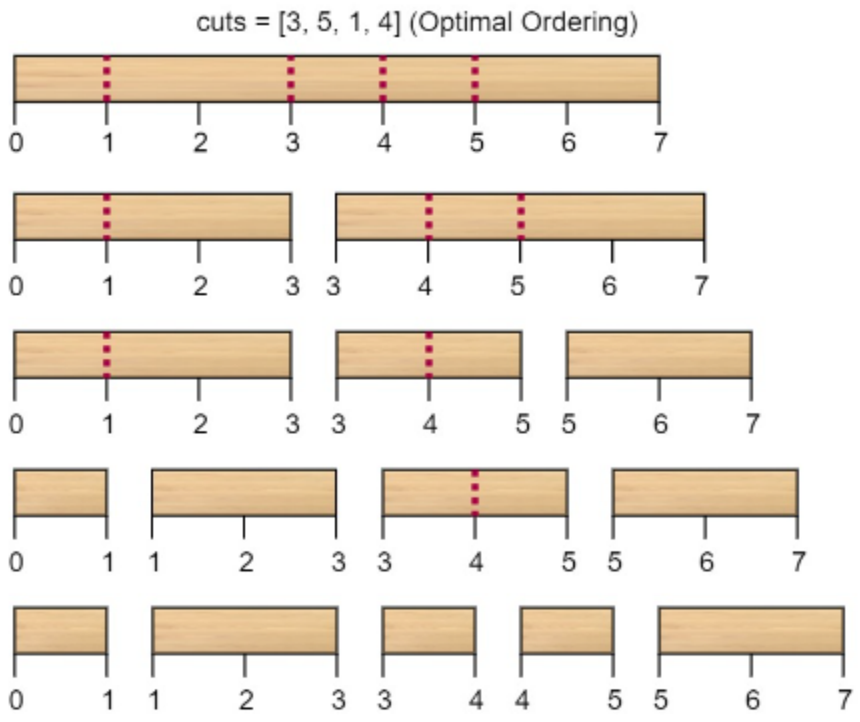
Return *the minimum total cost* of the cuts.

Example 1:

Input: n = 7, cuts = [1,3,4,5]
Output: 16
Explanation: Using cuts order = [1, 3, 4, 5] as in the input leads to the following scenario:



The first cut is done to a rod of length 7 so the cost is 7. The second cut is done to a rod of length 6 (i.e. the second part of the first cut), the third is done to a rod of length 4 and the last cut is to a rod of length 3. The total cost is 7 + 6 + 4 + 3 = 20.
Rearranging the cuts to be [3, 5, 1, 4] for example will lead to a scenario with total cost = 16 (as shown in the example photo 7 + 4 + 3 + 2 = 16).



Example 2:

```
Input: n = 9, cuts = [5,6,1,4,2]
Output: 22
Explanation: If you try the given cuts ordering the cost will be 25.
There are much ordering with total cost <= 25, for example, the order [4, 6, 5,
2, 1] has total cost = 22 which is the minimum possible.
```

Problem 2:

Given two strings word1 and word2, return *the minimum number of operations required to convert word1 to word2*.

You have the following three operations permitted on a word:

- Insert a character
- Delete a character
- Replace a character

Example 1:

```
Input: word1 = "horse", word2 = "ros"
Output: 3
Explanation:
horse -> rorse (replace 'h' with 'r')
rorse -> rose (remove 'r')
rose -> ros (remove 'e')
```

Example 2:

```
Input: word1 = "intention", word2 = "execution"
Output: 5
Explanation:
intention -> inention (remove 't')
inention -> enention (replace 'i' with 'e')
enention -> exention (replace 'n' with 'x')
exention -> exection (replace 'n' with 'c')
exection -> execution (insert 'u')
```

Problem 3:
Given a set of non-negative integers and a value **sum**, the task is to check if there is a subset of the given set whose sum is equal to the given **sum**. Return True if there is subset whose sum is our target sum, otherwise false.

Example 1:
Input: set[] = {3, 34, 4, 12, 5, 2}, sum = 9
Output: True
Explanation: There is a subset (4, 5) with sum 9.

Example 2:
Input: set[] = {3, 34, 4, 12, 5, 2}, sum = 30
Output: False
Explanation: There is no subset that add up to 30.