1449. Form Largest Integer With Digits That Add up to Target

Description

Given an array of integers `cost` and an integer `target`, return *the **maximum** integer you can paint under the following rules*:

- The cost of painting a digit `(i + 1)` is given by `cost[i]` (**0-indexed**).
- The total cost used must be equal to `target`.
- The integer does not have `0` digits.

Since the answer may be very large, return it as a string. If there is no way to paint any integer given the condition, return `"0"`.

Example 1:

```
Input: cost = [4,3,2,5,6,7,2,5,5], target = 9
Output: "7772"
Explanation: The cost to paint the digit '7' is 2, and the digit '2' is 3. Then cost("7772") = 2*3+ 3*1 = 9. You could also paint "977", but "7772" is the largest number.
Digit    cost
  1  ->   4
  2  ->   3
  3  ->   2
  4  ->   5
  5  ->   6
  6  ->   7
  7  ->   2
  8  ->   5
  9  ->   5
```

Example 2:

```
Input: cost = [7,6,5,5,5,6,8,7,8], target = 12
Output: "85"
Explanation: The cost to paint the digit '8' is 7, and the digit '5' is 5. Then cost("85") = 7 + 5 = 12.
```

Example 3:

```
Input: cost = [2,4,6,2,4,6,4,4,4], target = 5
Output: "0"
Explanation: It is impossible to paint any integer with total cost equal to target.
```

Constraints:

- `cost.length == 9`
- `1 <= cost[i], target <= 5000`