Dice from pennies

Colin likes to play role playing games. These games usually involve dice with varying numbers of sides; a 20 sided die is known as a D20 (marked 1 - 20 inclusive), a regular six sided die known as D6, and so on. Unfortunately, Colin has forgotten his bag of dice, and has only a jar full of pennies. Help him make his saving throw! For each required roll, Colin will tip out the pennies and read off whether they are heads or tails. This will serve as the random input to a function that returns a number fitting the range of the die. For example, to simulate a regular die (D6), the pairing is as follows:

```
HHH = 1
HHT = 2
HTH = 3
HTT = 4
THH = 5
THT = 6
```

and analogously for other dice using the minimum number of pennies to encode all the needed die numbers. See sample input for examples. Note that some outputs from the coins will not match to a die number when read from the first position. In this case, use the next coins in the list to return a result. For example, in case of 6 sided dice, TTTHHH maps to the value 1, not 5 as someone may expect.

In this problem you will implement two methods in the `DiceFromPennies` class. The first method is the `getRoll(String pennies)` method which calculates the numeric value of the roll given the `String pennies` representing the heads and tails. You may assume every die will have two or more sides and 32 sides or less.
That is, `2 <= numSides <=32`.

If the String of heads and tails does NOT generate a valid roll, return -1. For example, with a six sided die, the following Strings return -1: `"TTH"` and `"TTT"`.

The following code shows the results of the `getRoll` method with a six sided die.

The following code	Returns
`DiceFromPennies dfp = new DiceFromPennies(6);`	
`dfp.getRoll("HHH");`	1
`dfp.getRoll("HHT");`	2
`dfp.getRoll("HTH");`	3
`dfp.getRoll("HTT");`	4
`dfp.getRoll("THH");`	5
`dfp.getRoll("THT");`	6
`dfp.getRoll("TTT");`	-1
`dfp.getRoll("TTTHHH");`	1

This problem borrowed from a Saturday, October 14, 2006 programming contest: http://ugweb.cs.ualberta.ca/~acpc/
A joint effort of the University of Alberta, the University of Calgary, and the University of Lethbridge.

The following code shows the results of the `getRoll` method with a 20 sided die.

The following code	Returns
`dfp = new DiceFromPennies(20);`	
`dfp.getRoll("THHHTHHHHT");`	18
`dfp.getRoll("TTHHTHHTHT");`	6

The second method to complete is: `getRolls(int numDice, String pennies)`. This method calculates the numeric value of rolling one or more dice from the `String` representing the heads and tails.

You may assume every roll will have at least 1 die and will have 50 or fewer die (`1 <= numDice <= 50`). You may also assume that the `String pennies` will contain enough valid 'rolls'. That is, you can assume that `pennies` is long enough to get `numDice` valid rolls.

The following code shows the results of the `getRolls` method with a two sided die.

The following code	Returns
`dfp = new DiceFromPennies(2);`	
`dfp.getRolls(2, "HT");`	3
`dfp.getRolls(5, "HTTHHHTHTHTHTHTHT");`	7 =1+2+2+1+1

The following code shows the results of the `getRolls` method with a 15 sided die.

The following code	Returns
`dfp = new DiceFromPennies(15);`	
`dfp.getRolls(4, "HTTHHHTHTHTHTHTT");`	33

Note: *The above is the best description that Chris has come up with. Trying to sort out what he meant is part of the challenge.*