APPRENTICE TECHNICAL LOG DOC

D1000000 (GOOGLE JAM 2022)

Jorge Fernando Peña Vázquez

In this problem, we are given a collection of dice, each with a different number of sides. The i-th die has Si sides, and each side shows a different integer from 1 to Si. We need to choose some of the dice and pick one number from each die to form a straight. A straight is a sequence of integers that are consecutive, such as 3, 4, 5, 6. The length of a straight is the number of integers in the sequence.

The goal of the problem is to find the longest straight that we can form by picking one number from each of the chosen dice. Note that we can choose to use all of the dice, or we can choose to use only a subset of the dice. We need to find the longest straight that we can form in this way.

https://codingcompetitions.withgoogle.com/codejam/round/0000000000876ff1/0000000000a46471#problem

Overview:

To solve this problem, we can start by sorting the dice in increasing order of the number of sides. This will allow us to consider the dice in a logical order when trying to form a straight. We can then iterate through the dice and try to extend the current straight by adding the numbers on the current die to the straight. If the numbers on the current die can be added to the straight to form a longer straight, we can update the longest straight that we have found so far. Otherwise, we can move on to the next die and try to extend the straight using the numbers on that die.

Once we have considered all of the dice, we will have found the longest straight that we can form using the given collection of dice. We can then output the length of this straight as the solution to the problem.

Solution:

```
function maxDiceLength(inputStr){
  let length=0;
  let die=inputStr.split(" ")
  for(let i=0;i<die.length;++i){
    die[i]=Number(die[i]);
  }
  die=die.sort(function(a, b) {return a - b;});
  for(let i=0;i<die.length;++i){
    if(die[i]>length) ++length;
  }
  return length;
}
```

1. The function takes an input string as its only argument. This input string contains a space-separated list of the sizes of the dice in the collection.
2. The first line of the code creates a variable called **length** and initializes it to 0. This variable will be used to track the length of the longest straight that can be formed from the dice.
3. The next line of the code uses the **split()** method to split the input string into an array of strings, each representing the size of a single die. The resulting array is stored in a variable called **die**.
4. The code then iterates over the **die** array using a **for** loop. For each element in the array, the code converts it to a number using the **Number()** function and assigns the result back to the element in the array. This ensures that all elements in the **die** array are numbers, not strings.
5. The **die** array is then sorted in ascending order using the **sort()** method. This is necessary because the problem requires that the longest straight be formed from consecutive integers, and the dice must be selected in ascending order of their size in order to form the longest possible straight and makes finding a straight easier as every single straight will start with 1.
6. The code then iterates over the sorted **die** array using a **for** loop. For each element in the array, the code checks if the value of the element

is greater than the current value of the **length** variable. If this is the case, the code increments the value of the **length** variable by 1.

7. After the **for** loop has completed, the code returns the final value of the **length** variable, which represents the length of the longest straight that can be formed from the dice.

Alternative Solution(s):

One alternative approach to solving this problem would be to generate all possible permutations of the input dice and evaluate each permutation to find the longest straight that can be formed from it. However, this approach becomes computationally expensive and slow for large inputs, and is therefore not practical for most cases.