Solution to RollingDiceDivOne of SRM 536

stoundmire

March 11, 2012

1 Basic Observations

The goal of the problem is quite straightforward: given a set of dice indexed from 0 to $n-1$, each of which has dice$[i]$ faces, with $j+1$ pips on face j, you are to roll these dice and calculate the sum of pips on each face that comes out on top. Which sum has the maximum number of ways to reach? Give the smallest one in case of a tie.

First let's deal with some small test cases using *Brute Force*. Suppose we choose $\{3, 10, 3\}$, the number of ways to get the corresponding sums are given in table 1.

3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	3	6	8	9	9	9	9	9	9	8	6	3	1

Table 1: The number of ways to get the sums

Now we got:

- The results are symmetric.
- The results first increase, then stay the same, and at last decrease.

These will become obvious as we go along. Here we just assume they are right, and prove them as we derive our algorithms.

Then follow this case step by step to gain further insights. This is illustrated in table 2.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	3	3	3	3	3	3	3	2	1	0	0	0
1	3	6	9	9	9	9	9	9	9	9	9	9	6	3	1

Table 2: Step by step results

2 Analysis

2.1 First Approach

The previous observations may lead us to try to solve this problem by *Binary Search*. If there's an efficient way to figure out the number of ways to get a specific sum, then finding the first sum the number of ways to get which is not greater than its successor suffices. But unfortunately, the essential subproblem cannot be solved efficiently given the large values of dice.

2.2 Second Approach

Now let's tackle this problem by keeping track of the "boundaries" of the interval of sums which have the largest results. And for every new dice, update the "boundaries" accordingly. We will donate such intervals as $[l, r]$, the value dice$[i]$ as d_i(0-indexed) and the largest result as w.

2.2.1 Tips

The sums having nonzero results before processing dice $i(i \geq 1)$ form the interval $[i, \sum_{j=0}^{i-1} d_j]$(0-indexed). At first, the number of ways to get each sum in $[1, d_0]$ is exactly 1.

2.2.2 Situation I

Suppose that the length of the interval $[l, r]$ is no smaller than d_i. Now the number of ways to get one particular sum in the range $[l + d_i, r + 1]$ is $d_i \cdot w$, and $[l, r]$ is simply substituted by this new range.

2.2.3 Situation II

Suppose that the length of $[l, r]$ is smaller than d_i while the length of $[i, \sum_{j=0}^{i-1} d_j]$ is no smaller than d_i. Let $x = d_i - (r - l + 1)$, $[l, r]$ is substituted by $[l + d_i - \lfloor \frac{x+1}{2} \rfloor, r + 1 + \lfloor \frac{x+1}{2} \rfloor]$, due to the fact that the results before and after $[l, r]$ are **strictly** increasing and decreasing, respectively.

Let's look into this situation carefully. Suppose $d_i = 7$, the current interval with nonzero results is $[s_1, s_{12}]$ and the interval with maximum results is $[s_5, s_8]$. Define matrix $A_{7\times 12}$ such that $A_{ij} = i + s_j$. All elements on the same secondary diagonal are equal.[1] Define matrix $M_{7\times 12}$ such that M_{ij} is equal to the number of ways to get s_j. See figure 1 for details. In figure 1, $a < b < c < d < x$. The sum of elements on a secondary diagonal corresponds to the number of ways to get a particular sum of pips. To get the new interval with the maximum results, we have to choose the sums that use *all* current sums with the maximum number of ways to get and sums with slightly smaller number of ways to get on both the left and the right sides, the number of such sums on either side *differ by at most* 1. And Matrix M essentially proves the observations before.

[1] $s_k + 1 = s_{k-1} + 2 = \ldots$

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}
1	a	b	c	d	x	x	x	x	d	c	b	a
2	a	b	c	d	x	x	x	x	d	c	b	a
3	a	b	c	d	x	x	x	x	d	c	b	a
4	a	b	c	d	x	x	x	x	d	c	b	a
5	a	b	c	d	x	x	x	x	d	c	b	a
6	a	b	c	d	x	x	x	x	d	c	b	a
7	a	b	c	d	x	x	x	x	d	c	b	a

Figure 1: Matrix M

2.2.4 Situation III

Suppose that the length of the interval $[l, r]$ and the length of the interval $[i, \sum_{j=0}^{i-1} d_j]$ are both smaller than d_i. Now the number of ways to get one particular sum in the range $[(\sum_{j=0}^{i-1} d_j - i + 1) + i, d_i + i]$ is the sum of all previous nonzero results, and $[l, r]$ is simply substituted by this new range.

3 Some Optimization

Sorting dice from largest to smallest eliminates situation III.