0.1 Generating Partitions

- *Algorithm:* name (algo. 1)
- *Input:* An positive integer n
- *Complexity:* $\mathcal{O}(n^2)$
- *Data structure compatibility:* N/A
- *Common applications:* The Durfee square can be used to prove many partition identities in the field of combinatorics.

Problem. Generating Partitions

Given an positive integer n, we want to find the number of different tuples $(a_1, a_2, ..., a_k)$, such that two constrains are satisfied. Firstly, $a_1 + a_2 + ..., a_k = n$. Secondly, $a_1 \geq a_2 \geq ... \geq a_k > 0$[4].

Description

A first look at the problem

Given an positive integer n, we want to find the number of different tuples $(a_1, a_2, ..., a_k)$, such that two constrains are satisfied. Firstly, $a_1 + a_2 + ..., a_k = n$. Secondly, $a_1 \geq a_2 \geq ... \geq a_k > 0$. Some variation of this problem can be find by setting additional constrains like k is even number. We define a partition function $p(n)$ as the number of tuples we find in the problem. In this paper, we will focus on the partition problem without additional constrins to make our life easier. For example, we can find out there are only one partition (1) for $n = 1$. For $n = 5$, there are 7 partitions, (1, 1, 1, 1, 1), (1, 1, 1, 2), (1, 1, 3), (1, 4), (1, 2, 2), (5), (2, 3). Here we define part as the summand in each partition, for example $1 + 1 + 3$ is the part for tuple (1, 1, 3). The order of elements in the tuples does not matter in our problem.

The idea of this algorithm

To save time for future calculation, we use dynamic programming to store the values used in former calculation. Notice that in the for loop, for example, we can use the value of array[j - i]. This is quite similar to Bellman-ford that we learned in chapter 4 of the slides[2].

Time complexity

In this algorithm, there are two levels of for loops, the time complexity is $\mathcal{O}(n) \times \mathcal{O}(n) = \mathcal{O}(n^2)$, the first loop to set the value of array does not matter. Comparing to the size of input, which is $\mathcal{O}(\log n)$, the time complexity would be exponential. Some approximation algorithms are proposed to obtain the value with less accuracy within polynomial time. Also we can directly calculate the asymptotics of the number by

$$\log p(n) \sim C\sqrt{n}$$

where C is a constant and $C = \pi\sqrt{\frac{2}{3}}$[1].

Or we can express the formula in another way as

$$p(n) \sim \frac{1}{4n\sqrt{3}} \exp\left(\pi\sqrt{\frac{2n}{3}}\right)$$

.

Another way to tackle this problem

We can use generating function to calculate the value as well[3].

$$\sum_{n=0}^{\infty} q(n)x^n = \prod_{k=1}^{\infty}\left(1+x^k\right) = \prod_{k=1}^{\infty}\frac{1}{1-x^{2k-1}}$$

, by using this equation, we can get the number of partition by writing down the coefficient before the term x^n. For example, to find the number of partitions when $n = 8$, we need to calculate

$$\begin{gathered}\left(1+x+x^2+x^3+x^4+x^5+x^6+x^7+x^8\right)\left(1+x^2+x^4+x^6+x^8\right)\left(1+x^3+x^6\right)\\ \left(1+x^4+x^8\right)\left(1+x^5\right)\left(1+x^6\right)\left(1+x^7\right)\left(1+x^8\right)\\ = 1+x+2x^2+3x^3+5x^4+7x^5+11x^6+15x^7+22x^8+\cdots+x^{56}\end{gathered}$$

Then we can get the number by reading the coefficient of x^8, which is 22.

Further application

We can define the rank of a certain partition of the positive integer n as the largest integer k so that we have at least k numbers in partition that is greater or equal to k. Furthermore, two diagrams, namely Ferrers diagram and Young diagram are used for visualization representation of this kind of problem. Durfee square can be defined as the $k \times k$*squarecountingfromthelargestelementinthepartition*, where k stands for the rank we give.

Algorithm 1: Partition

Input : a positive integer n
Output: the number of partition, where partition is defined in problem definition

```
int array[n] for inti = 1; i <= n; i++ do
    array[i] = 1
end for
for inti = 1; i <= n; i++ do
    for intj = i; j <= n; j++ do
        array[j] = array[j] + array[j - i]
    end for
end for
return array[n]
```

References.

[1] George E Andrews. *The theory of partitions*. 2. Cambridge university press, 1998 (cit. on p. 1).

[2] Manuel. *VE477 – Introduction to Algorithms (lecture slides)*. 2020 (cit. on p. 1).

[3] whitman.edu. *Partitions of Integers*. [Online; accessed 5-October-2020]. 2020. URL: https://www.whitman.edu/mathematics/cgt_online/book/section03.03.html (cit. on p. 2).

[4] Wikipedia contributors. *Partition (number theory)*. [Online; accessed 5-October-2020]. 2020. URL: https://en.wikipedia.org/wiki/Partition_(number_theory) (cit. on p. 1).