

Counting Partitions - Alan Yan

Define the following functions

$A(n, k)$ = the number of partitions of n with largest part k and an even number of parts.

$B(n, k)$ = the number of partitions of n with largest part k and an odd number of parts.

Then, we have that

$$A(n, k) = A(n-1, k-1) + B(n-k, k)$$

$$B(n, k) = B(n-1, k-1) + A(n-k, k)$$

The rest is dynamic programming. My code is given below.

```

A = {}
B = {}
p = 10**9 + 7

r = open("file.txt", "r")
text = r.read()
text = text[:len(text)-1]
a_list = [int(i.split(" ")[0]) for i in text.split("\n")]
b_list = [int(i.split(" ")[1]) for i in text.split("\n")]
a = max(a_list)
A[(1,1)] = 0
B[(1,1)] = 1
for n in range(1, a+1):
    for k in range(n+1):
        if(n == k):
            A[(n,k)] = 0
            B[(n,k)] = 1
        elif(k == 0):
            A[(n,k)] = 0
            B[(n,k)] = 0
        else:
            A[(n,k)] = A[(n-1, k-1)]
            B[(n,k)] = B[(n-1, k-1)]
            if(n-k >= k):
                A[(n,k)] += B[(n-k, k)]
                B[(n,k)] += A[(n-k, k)]

tot = 0
for i in range(len(a_list)):
    tot += B[(a_list[i],b_list[i])]
print(tot)

```