

# Proyecto Final

## Matemática Discreta

### Resumen:

En teoría de números una partición de un número entero  $N$  es una forma de descomponer este número en una suma de enteros menores a  $N$ , que sumados son iguales a  $N$ . La longitud  $k$  de una partición está definida por la cantidad de sumandos que conforman a la partición. Por ejemplo las particiones del número 5 son:

$4 + 1$	$k = 2$
$3 + 1 + 1$	$k = 3$
$3 + 2$	$k = 2$
$2 + 1 + 1 + 1$	$k = 4$
$2 + 2 + 1$	$k = 3$
$1 + 1 + 1 + 1 + 1$	$k = 5$

(Andrews, 1976)

### Metodología:

Una de las muchas maneras de obtener las particiones de un número entero es obteniendo todas las maneras de elegir  $k$  elementos del conjunto  $A$  permitiendo repetición generando un conjunto  $B$  formado por subconjuntos.

Donde:

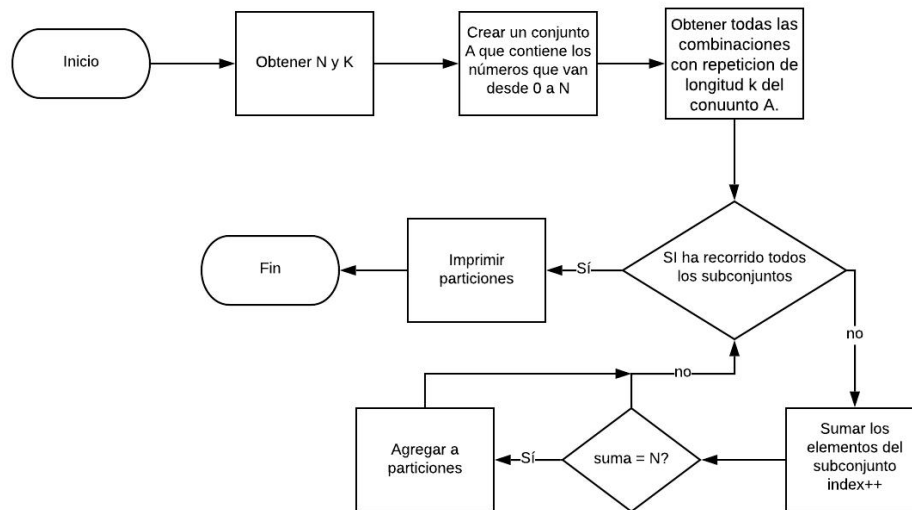
$A = \{1, 2, 3, 4, 5 \dots N\}$  y

$B =$  Combinatorias con repetición de  $k$  elementos de  $A$

En otras palabras eligiendo todas las combinaciones con repetición de longitud  $k$  del conjunto  $A$  se puede sumar cada elemento de los subconjuntos formados para descartar aquellos cuya suma no sea igual a  $N$ . Se escogen combinaciones ya que esto hace que sea un proceso mucho más fácil ya que permite que se eviten que se obtengan dos particiones pero con los sumandos en orden diferente, que en teoría de números representan la misma partición, ya que dos sumas se considerarán iguales si solo difieren en el orden de los sumandos.

## Diagrama de Flujo:

La metodología anteriormente descrita se puede escribir utilizando un diagrama de flujo como el que se muestra a continuación:



## Descripción de la Solución:

Para poder implementar esta metodología en un programa en el interprete de python 3 se hizo uso de la función `combinations_with_replacement()` de la libreria `itertools`, para poder obtener todas las combinatorias con repeticion estas fueran almacenadas en listas y se escribió una función que cmple esta utilizada. También se escribió otra función que permite parsear estas particiones en un string donde los elementos se separan por medio de un signo "+", luego se obtuvieron todas aquellas que eran particiones, y si imprimieron los resultados. Pero este algoritmo tiene una desventaja cuando el valor de K toma un valor grande debido a que el algoritmo empleado para generar las combinatorias es complejo, es por eso que se ha solucionado la interfaz con el usuario colocando un "Processing Data... Wait" .

## Referencias:

Andrews, George E. (1976). *The theory of partitions (1st pbk. ed edición)*. Cambridge University Press. p. 63. ISBN 9780521637664. OCLC 39742738.  
Python 3 (2019) `itertools` Documentations Información obtenida de <https://docs.python.org/3/library/itertools.html>  
Link del repositorio <https://github.com/suulcoder/PartionsGetter>