# Tarea 6- Programación Avanzada

Daniel Vallejo Aldana
`daniel.vallejo@cimat.mx`

28 de abril de 2021

**Problema 1.** Implementación de la clase Big Int

**Solución** Regerirse al archivo `Bigint.h` en donde se encuentran implementadas las funciones de suma,resta multiplicación, módulo, división. La función de multiplicación se implementó con la multiplicación de Karatsuba. Las funciones se prueban en el archivo `problema 1.cpp`

**Problema 2.** Dado un tablero de $n \times n$, ¿De cuántas formas se pueden colocar dos reinas antagonistas tal que siempre se amenacen?

**Solución** Consideremos primero el número de formas en las que podemos colocar a las dos reinas antagonistas independientemente si se amenazan o no. Notemos que el número de posiciones en las que podemos colocar a la primera reina es $n^2$, como una posición ya está ocupada entonces a la segunda reina la podemos colocar en $n^2 - 1$ lugares. Entonces al haber dos reinas el número posible de posiciones en las que podemos poner a las dos reinas es

$$P(n) = \frac{n^2(n^2-1)}{2}$$

Sabemos además que dos reinas antagonistas se amenazan si se encuentran en la misma fila, en la misma columna o en alguna de las diagonales, lo anterior nos dice que las posiciones en las cuales podemos poner las reinas sin que se amenacen es

$$N(n) = n(n-1)(n-2)(3n-1)/6$$

Es decir no colocarla en la fila, columna o diagonal de la reina antagonista.

Realizando la resta de estas dos cantidades obtenemos que el numero de posiciones para las cuales dos reinas se amenazan es

$$A(n) = P(n) - N(n) = n(5n-1)(n-1)/3$$

**Problema 3.** Consideramos otra vez un tablero de ajedrez de tamaño $p \times p$ $(0 \leq p \leq 300)$ y de lado unitario. Si un rey quiere hacer caminos en el tablero pasando una y sola vez por cada celda, regresando a su punto inicial, sin cruzar su camino (excepto al final), ¿cual será el tamaño del camino más largo (en distancia Euclideana en el tablero) que podria hacer?

**Solución** Para la solución de este problema podemos notar que la distancia del tour del rey se maximiza hacindo recorridos en diagonal ya que la distancia de estos es $\sqrt{2} > 1$, entonces para calcular la distancia máxima debemos de encontrar el máximo número de diagonales que podemos hacer en un recorrido para un tablero de ajedréz de tamaño $n \times n$ con $n \in [0, 300]$. De acuerdo a simulaciones realizadas para tamaños de $2 \leq n \leq 7$, se puede ver que a lo más hay $(n-2)^2$ diagonales en el recorrido por lo que la máxima distancia está dada por

$$MaxDist = (n^2 - (n-2)^2) + (n-2)^2\sqrt{2}$$

**Problema 4.** Para un polinomio multivariado

$$(x_1, .., x_k)^n$$

cuyos monomios son de la forma $x_1^{i_1}...x_k^{i_k}$ Escribir un programa que dada la secuencia de los $i_k$, $n$ y $k$ regrese el coeficiente del monomio

**Solución** Notemos que el coeficiente de el monomio $x_1^{i_1}...x_k^{i_k}$, está dado por

$$\frac{n!}{i_1!...i_k!}$$

Ver archivo `problema 4.cpp` para implementación

**Problema 5.** Determinar para $n \geq 0$, el número más grande de Kaprekar $k$, tal que $k \leq n$ si no existe se regresa $-1$

**Solución** Revisar `problema 5.cpp`