



TA1

parte 1

1. se genera creando un nuevo vector del tamaño máximo indicado, luego en el for se completa cada item del vector con `i` que comienza en 0 y se va actualizando en incrementos de 1 con cada iteración para generar números consecutivos en orden ascendente
2. se genera creando un nuevo vector del tamaño máximo indicado, luego en el for se completa cada item del vector con `tamañoMax - i`, `i` comienza en 0 y en cada iteración incrementa en 1, por lo que el primer elemento sería `tamañoMax - 0` (= tamañoMax), la siguiente `tamañoMax - 1`, y así sucesivamente hasta completar el vector generando un vector de números consecutivos en orden descendente
3. se crean dos vectores ambos del tamaño máximo especificado (`m`), uno de enteros (`datosGenerados`) y otro de booleanos (`datosUtilizados`)
en el bucle for se genera un num aleatorio (`j`) entre 0 y el largo del vector, si esta posición está marcada con `true` en el vector de datos utilizados, se recorren las posiciones siguientes (consecutivas) hasta encontrar un espacio vacío, `j` se va actualizando para que el encontrar un espacio vacío esté almacenando el índice de esa posición
luego en la posición `j` de datos generados se almacena el número de iteración (la iteración es un número dentro del rango que queremos generar, y la posición `j` es aleatoria), y en la posición `j` de datos utilizados se actualiza a `true` para indicar que ya se puso ese número en el array de datos generados
finalmente se devuelve el vector de datos generados

no pueden existir claves repetidas porque el vector de `datosUtilizados` indica qué números se generaron, si la posición marca `true` `j` se actualiza hasta encontrar un número que no esté puesto en el vector de números generados, por lo que no se repiten las claves

el orden de tiempo de ejecución es $O(n^2)$ porque el for itera `m` veces y en el peor de los casos si todas las entradas están ocupadas menos la anterior a la `j` actual, hay que recorrer todo el vector para encontrar la posición vacía
4. tiene 32000 elementos (tamaño máximo), para parametrizar la cantidad de datos en vez de declarar el tamaño máximo en la clase, se puede pedir un entero por parámetro y que sea el tamaño máximo que se usa
5. hacemos un método que verifique por cada posición del vector si el que está en la posición siguiente es mayor o igual, si es falso devuelve falso, de lo contrario devuelve verdadero es de $O(n)$ porque recorre todas las posiciones del vector