3. Ejercicios obligatorios de la práctica

IMPORTANTE: Los ejercicios de esta sección son de carácter individual y serán calificados. Cada ejercicio podrá estar aprobado, o será devuelto para incorporar correcciones. Las consultas sobre estos ejercicios deben limitarse a expresar dudas de interpretación sobre el enunciado. Además, como el objetivo de estos ejercicios es evaluar el desempeño individual, no se permite trabajar grupalmente ni compartir soluciones. La fecha límite de entrega es la fecha de finalización del bloque.

Nota (★): en los ejercicios que siguen, trabajaremos con conjuntos de naturales, dados por un módulo ConjuntoDeNaturales. Para recorrer los conjuntos, se puede suponer que el módulo ConjuntoDeNaturales cuenta con un iterador que visita los elementos en orden estrictamente creciente, con tiempo constante de creación, acceso y avance. Por ejemplo, si C es un conjunto de cardinal n, el siguiente `for` muestra los elementos del conjunto C ordenados de menor a mayor y tiene costo lineal en peor caso, es decir, cuesta $O(n)$:

```
for x in C {
    mostrar(x)
}
```

Se puede escribir más explícitamente con un `while`:

```
it := CrearIt(C)              // CrearIt       cuesta O(1)
while HaySiguiente?(it) {     // HaySiguiente? cuesta O(1)
    mostrar(Siguiente(it))    // Siguiente     cuesta O(1)
    Avanzar(it)               // Avanzar       cuesta O(1)
}
```

No se pueden asumir otras operaciones en la interfaz del módulo ConjuntoDeNaturales. En particular, dicho módulo **no** provee una operación para calcular la intersección.

3.1. Ordenamiento

Ejercicio 1: ordenamiento

Se tiene un arreglo A de N conjuntos $A[1], \ldots, A[N]$. El cardinal de cada conjunto es **a lo sumo** K, es decir $\#(A[i]) \leq K$ para todo $1 \leq i \leq N$. Se desea ordenar el arreglo A para obtener como resultado un arreglo de conjuntos $B[1], \ldots, B[N]$ de tal modo que se cumpla la siguiente condición:

$$B[i] \subseteq B[j] \;\Rightarrow\; i \leq j \qquad \text{para todo } i, j \text{ en el rango } 1..N$$

es decir, si un conjunto está incluido en otro, debe aparecer antes en el arreglo ordenado.

Notar que el problema no tiene única solución. Por ejemplo si A es el siguiente arreglo de conjuntos:

$$A \;=\; [\{1,2,3\}, \{1,2\}, \{1,3\}, \{1,2,4\}]$$

Una posible respuesta podría ser el siguiente arreglo B:

$$B \;=\; [\{1,3\}, \{1,2\}, \{1,2,3\}, \{1,2,4\}]$$

y otra posible respuesta podría ser el siguiente arreglo B':

$$B' \;=\; [\{1,2\}, \{1,2,4\}, \{1,3\}, \{1,2,3\}]$$

Proponer un algoritmo para resolver este problema con complejidad temporal en peor caso $O(NK)$, teniendo en cuenta que no hay conjuntos repetidos. Justificar la complejidad temporal obtenida. Para manipular los conjuntos tener en cuenta la **nota** (★) de arriba.

3.2. Dividir y conquistar

Ejercicio 2: dividir y conquistar

Se tiene un arreglo C de N conjuntos $C[1], \ldots, C[N]$. El cardinal de cada conjunto es **exactamente** M, es decir $\#(C[i]) = M$ para todo $1 \leq i \leq N$.

1. Proponer un algoritmo para determinar si los conjuntos $C[1], \ldots, C[N]$ son **disjuntos dos a dos**, es decir si para todo $i \neq j$ en el rango $1..N$ se tiene que $C[i] \cap C[j] = \emptyset$. La complejidad temporal en peor caso del algoritmo debe ser $O(MN \log N)$. Justificar la complejidad temporal obtenida.

2. (**optativo**) Suponiendo que los elementos de todos los conjuntos están en el rango comprendido entre 1 y el producto $M \cdot N$, proponer un algoritmo para resolver el mismo problema con complejidad temporal en peor caso $O(MN)$. Notar que este algoritmo sirve para determinar si los conjuntos $C[1], \ldots, C[N]$ constituyen una **partición** del conjunto $\{1, 2, \ldots, MN - 1, MN\}$.
   Justificar la complejidad temporal obtenida.

Para manipular los conjuntos tener en cuenta la **nota** (★) de arriba.

Aclaración: el ejercicio que define la aprobación es el 2.1, el que está marcado como optativo define si, de estar aprobado el anterior, se sube la nota para que quede una P en lugar de una A.
De estar mal resuelto el 2.1, la nota final sería de una R o I independientemente del 2.2.