

1158065 - Laboratorio de Optimización  
Tarea 4 - 2025-I.

Sea  $U = \{0, 1, \dots, n-1\}$  un conjunto universo y sea  $C = (C_0, C_1, \dots, C_{m-1})$  una colección de  $m$  subconjuntos de  $U$ , donde  $|C_i| = t$  y además  $\bigcup C_i = U$ . El problema de cobertura mínima de conjuntos (*minimum set cover* en inglés) consiste en elegir la menor cantidad de conjuntos  $C_i$  tales que su unión sea  $U$ .

Una manera de modelar el problema es la siguiente. Sea  $I = \{0, 1, \dots, m-1\}$  y sea  $x_i$  una variable binaria que indique si el subconjunto  $C_i$  debe elegirse.

- Función objetivo que minimiza el número de subconjuntos elegidos.

$$\text{minimizar } \sum_{i \in I} x_i$$

- Restricciones que obligan a cubrir cada elemento de  $U$ .

$$\sum_{i \in I: v \in C_i} x_i \geq 1 \quad \text{para } 0 \leq v \leq n-1$$

El siguiente algoritmo glotón es un algoritmo de aproximación de factor  $\log_2(n)$ . Sea  $V$  el conjunto de los elementos ya cubiertos de  $U$  y sea  $R$  la colección de índices de los subconjuntos  $C_i$  elegidos. La idea es elegir de forma glotona el subconjunto  $C_i$  que cubra más elementos aún no cubiertos, repitiendo el proceso hasta cubrir todo  $U$ . En caso de empate, elegiremos el subconjunto de menor  $i$ .

---

**subrutina** SETCOVERAPROXIMACIÓN( $n, m, t, C = (C_0, C_1, \dots, C_{m-1})$ )

```
V ← ∅, R ← ∅
mientras V ≠ U
  S ← ∅
  k ← -1
  para i ← 0 ... m - 1
    si |S| < |Ci \ V|
      S ← Ci \ V
      k ← i
  V ← V ∪ S
  R ← R ∪ k
regresa R
```

---

Escribe un programa que tome una instancia del problema anterior y que asista a Gurobi para calcular la solución óptima usando su API de programación. Antes de comenzar el proceso de optimización de Gurobi, tu programa deberá proporcionarle una solución inicial calculada con el algoritmo descrito arriba.

El formato de la entrada es el siguiente. La primera línea contiene el entero  $n$ , la segunda el entero  $m$  y la tercera el entero  $t$ . Cada una de las siguientes  $m$  líneas contiene  $t$  enteros separados por espacios que denotan los  $t$  elementos

de un subconjunto  $C_i$ . Puedes suponer que  $1 \leq n, m, t \leq 200$ . La salida debe consistir de dos líneas: la primera debe contener una secuencia ordenada de enteros separados por espacios que identifican los subconjuntos elegidos por el algoritmo de aproximación. La segunda debe contener una secuencia ordenada de enteros con identificadores de los subconjuntos elegidos en la solución óptima. En caso de existir más de una solución óptima, puedes imprimir cualquiera de ellas. Tu programa no debe imprimir nada adicional a lo solicitado, con la posible excepción del mensaje emitido por Gurobi sobre el uso académico de su licencia.

Deberás enviar a [racc@azc.uam.mx](mailto:racc@azc.uam.mx) el código fuente de tu programa con el nombre `matriculat4.cpp`. Tu programa deberá funcionar también para instancias distintas a la del ejemplo, pero cuya entrada siga el mismo formato.

Ejemplo de entrada	Ejemplo de salida
6	0 1 2
4	1 2
3	
0 3 5	
1 4 5	
0 2 3	
0 2 5	