

## EJERCICIO: COVERSET (conjuntos)

VIDEO EXPLICATIVO [VIDEO](#)

**INTRO** Queremos determinar si la unión de una serie de conjuntos `list<set<int>> LA` logra *cubrir* (o sea incluye) a algún conjunto la lista `list<set<int>> LB`. Para eso tomamos cada conjunto de `LB` y vamos tomando la unión de los primeros  $m$  conjuntos de `LA` hasta ver si para algún  $m$  la unión `UA` de los conjuntos de `LA` cubre a `B`. Si ninguno lo cubre, pasamos al siguiente conjunto de `LB` y así siguiendo.

**CONSIGNA** Escribir una función

```
void coverset(list<set<int>> &LA, list<set<int>> &LB, set<int> &UA, set<int> &B);
```

que realiza la tarea indicada. Es decir para cada conjunto `B` en `LB` verifica si la unión de una cierta cantidad de conjuntos de `LA` cubre a `B`. En caso de que así ocurra retorna por `UA` la unión correspondiente de los conjuntos de `A` y por `B` el conjunto de `LB` que es cubierto.

### EJEMPLOS

```
[Ej1] LA: [[1,2],[3,4],[5,6],[7,8]], LB: [[12,13],[11,12],[0,1],[1,2,3,4,5],[1,2,3]],  
      => B: [1,2,3,4,5], UA: [1,2,3,4,5,6]  
[Ej2] LA: [[1],[3,4],[5],[8,9]], LB: [[1,7],[2,5],[1,5],[1,2,3]]  
      => B: [1,5], UA: [1,3,4,5]  
[Ej3] LA: [[1],[3,4],[5],[8,9]], LB: [[1,7],[2,5],[1,2,5,9],[1,2,3]]  
      => B: [], UA: []
```

### AYUDA

- Hacer un lazo externo que recorre los conjuntos `B` de `LB`
- Para cada `B` ir haciendo la unión `UA` de los conjuntos de `A` y verificar si cubre a `B`.
- **NOTA:** Para ver si un conjunto `UA` cubre (o sea incluye) a otro `B`, basta con hacer la diferencia `B-UA` y chequear si está vacía. Si está vacía es porque `UA` cubre `B`.

**RESTRICCIONES SOLO USAR LAS FUNCIONES BINARIAS**

**ZIP** [Enlace al zip](#)