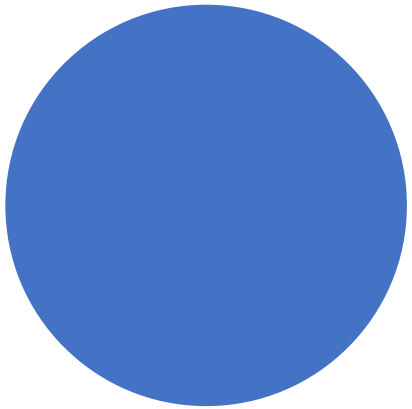
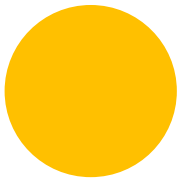

Algoritmos y Programación

Práctica 10a

Programación con Restricciones

Minizinc: count.mzn

- ¿Cuántos soldados tiene mi ejército? Sé que el número está entre 100 y 500.
  - Se ordenan en columnas de 5 soldados, y sobran 2
  - Se ordenan en columnas de 7 soldados y sobran 2
  - Se ordenan en columnas de 12 soldados, y sobra 1

  ¿Cuántos soldados hay?

- Variable de decisión:
  - army
- 3 restricciones

El formato de salida debe ser:

army = ***valor***

Minizinc:

- **solve satisfy**
- **mod**

Minizinc: army.mzn

- Reclutar un ejército
  - Presupuesto máximo 10000$
  - Soldados de 4 pueblos distintos (F,L,Z,J)
    - [F]: Fuerza:6. Coste: 13$. Máximo 1000 soldados
    - [L]: Fuerza:10. Coste 21$. Máximo 400 soldados
    - [Z]: Fuerza: 8. Coste 17$. Máximo 500 soldados
    - [J]: Fuerza:40. Coste 100$. Máximo 150 soldados
  - Maximizar la fuerza del ejercito

El formato de salida debe ser:

F = ***valor***
L = ***valor***
Z = ***valor***
J = ***valor***

Minizinc:

- **solve maximize**

Minizinc: sequence.mzn

- Secuencia
  - Construir un modelo sequence.mzn:
    - El parámetro n define la longitud del array
    - El array contiene variables de decisión con valores entre 0..3
    - Restricción: el primer valor es 0, el ultimo vale 3, y la suma de 2 números adyacentes en el array es como mucho 3
    - Restricción: el valor en posiciones divisibles por 3 tiene que ser mayor o igual a 2.
    - Maximizar la suma de los valores del array
    - La salida debe ser *suma = array de valores*
      - Por ejemplo***, 6 = [0,1,2,0,3]***
  - [Opcional] Ejecutarlo con valores de n entre 3 y 9

El formato de salida debe ser:

x = [***valores de x***]

Minizinc:

- **forall (… where …)**

Minizinc: Problema del ladrón

- Nuestro ladrón debe elegir qué casas debe robar para conseguir el máximo beneficio.
- Para que no se activen las alarmas si roba en una casa no puede robar en la siguiente.

Ejemplo:

N=5;
value=[3,10,3,1,2];

El formato de salida debe ser:

taken = [0, 1, 0, 0, 1]
Total Value = 12

Minizinc: Knapsack 0/1

- Plantear un modelo de la mochila.
  - Parámetros:
    - Crear un set of int de ITEMS
    - 2 arrays: value y weight
  - Variable de decisión:
    - Array: taken (indica el número de veces que un ítem se mete en la mochila)

Ejemplo:

capacity=10;
value=[45,48,35];
weight=[5,8,3];

El formato de salida debe ser:

taken = [1, 0, 1]
Total Value = 80

Minizinc: Knapsack con repetición

- Plantear un modelo de la mochila, donde los objetos pueden incluirse en la mochila 2 veces.
  - Parámetros:
    - Crear un set of int de ITEMS
    - 2 Arrays: value y weight
  - Variable de decisión:
    - Array: taken (indica el número de veces que un ítem se mete en la mochila)

Ejemplo:

capacity=10;
value=[45,48,35];
weight=[5,8,3];

El formato de salida debe ser:

taken = [2, 0, 0]
Total Value = 90