- Describa cuál es el criterio de selección.
- ¿En qué estructuras de datos representará la información del problema?
- Explique el algoritmo, es decir, los pasos a seguir para obtener el resultado. No se pide que "lea" el algoritmo ("se define una variable x", "se declara un for"), si no que lo explique ("se recorre la lista/el arreglo/" o "se elije de tal conjunto el que satisface...").
- Escriba el algoritmo en el lenguaje de programación de la materia.

9. (*sobredosis de limonada*) Es viernes a las 18 y usted tiene ganas de tomar limonada con sus amigos. Hay $n$ bares cerca, donde cada bar $i$ tiene un precio $P_i$ de la pinta de limonada y un horario de happy hour $H_i$, medido en horas a partir de las 18 (por ejemplo, si el happy hour del bar $i$ es hasta las 19, entonces $H_i = 1$), en el cual la pinta costará un 50% menos. Usted toma una cantidad fija de 2 pintas por hora y no se considera el tiempo de moverse de un bar a otro. Se desea obtener el menor dinero posible que usted puede gastar para tomar limonada desde las 18 hasta las 02 am (es decir que usted tomará 16 pintas) eligiendo en cada hora el bar que más le convenga.

   - **Criterio de selección**
     Elige el bar con la limonada más barata a cierta hora.

   - **Estructuras de datos**
     Planteo a los bares como una tupla de tres elementos que indica el nombre del local, el precio de la limonada y la cantidad de horas que hay de happy hour.

```
type Bar = tuple
             id: string
             precio: float
             happy_hour: nat
           end tuple

{- PRE: n ≥ 8 -}
fun sobredosis_de_limonada(B: Set of Bar) ret res: float
```

     Los bares que están abiertos son ordenados en un conjunto, y a medida que avanza la noche se va acumulando la cantidad de dinero.

   - **Descripción de cómo se soluciona el problema**
     En cada hora revisa el bar con el precio de la limonada más barato, teniendo en cuenta también los happy hour disponibles. Una vez que toma las dos pintas descarta el bar y busca otro diferente, repitiendo el procedimiento hasta que lleguen las 02:00 (o pasen las 8 horas).

   - **Definición del algoritmo**

```
type Bar = tuple
             id: string
             precio: float
             happy_hour: nat
           end tuple

{- PRE: n ≥ 8 -}
fun sobredosis_de_limonada(B: Set of Bar) ret res: float
    var B_aux: Set of Bar
    var b: Bar
    var hora: nat

    B_aux := copy_set(B)
    hora := 0
    res := 0
```

```
    while hora < 8 do
        b := elegir_bar(B_aux,hora)
        if b.happy_hour > hora then
            res := res + b.precio
        else
            res := res + (2 * b.precio)
        fi
        elim(B_aux,b)
        hora := hora + 1
    od

    destroy_set(B_aux)
end fun

fun elegir_bar(B: Set of Bar, h: nat) ret res: Bar
    var B_aux: Set of Bar
    var b: Bar
    var limonada_barata: float
    var precio_de_hora: float

    B_aux := copy_set(B)
    limonada_barata := ∞

    while not is_empty_set(B_aux) do
        b := get(B_aux)

        if b.happy_hour ≤ h then
            precio_de_hora := 2 * b.precio
        else
            precio_de_hora := b.precio
        fi

        if precio_de_hora ≤ limonada_barata then
            res := b
            limonada_barata := precio_de_hora
        fi

        elim(B_aux,b)
    od

    destroy_set(B_aux)
end fun
```