

- 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 elige de tal conjunto el que satisface...").
- Escriba el algoritmo en el lenguaje de programación de la materia.

8. Usted vive en la montaña, es invierno, y hace mucho frío. Son las 10 de la noche. Tiene una voraz estufa a leña y n troncos de distintas clases de madera. Todos los troncos son del mismo tamaño y en la estufa entra solo uno por vez. Cada tronco i es capaz de irradiar una temperatura k_i mientras se quema, y dura una cantidad t_i de minutos encendido dentro de la estufa. Se requiere encontrar el orden en que se utilizarán la menor cantidad posible de troncos a quemar entre las 22 y las 12 hs del día siguiente, asegurando que entre las 22 y las 6 la estufa irradie constantemente una temperatura no menor a $K1$; y entre las 6 y las 12 am, una temperatura no menor a $K2$.

- **Criterio de selección**

a) Entre las 22:00 y las 6:00 → elige el tronco con mayor tiempo t_i de encendido, cuya temperatura irradiada k_i es mayor o igual a $K1$.

b) Entre las 6:00 y las 12:00 → elige el tronco con mayor tiempo t_i de encendido, cuya temperatura irradiada k_i es mayor o igual a $K2$.

- **Estructuras de datos**

Planteo a los troncos como una tupla de tres elementos, contando con el identificador del tronco, la temperatura máxima que irradia mientras se quema y el tiempo que permanece quemándose.

```
type Troncos = tuple
    id: nat
    temperatura: float
    tiempo: nat
end tuple
```

```
fun estufa_voraz(T: Set of Troncos, K1: float, K2: float) ret res: List of Troncos
```

Los troncos disponibles están ordenados en un conjunto T , mientras que las temperaturas máximas que se esperan, $K1$ y $K2$, son variables de tipo float. Luego los troncos utilizados se derivan a una lista aparte.

- **Descripción de cómo se soluciona el problema**

Busca y elige el tronco que va a durar mayor tiempo quemándose pero que también la temperatura supere o iguale a $K1$ o $K2$, y así sucesivamente hasta que se supere el rango de las 14 horas u 840 minutos.

- **Definición del algoritmo**

```
type Troncos = tuple
    id: nat
    temperatura: float
    tiempo: nat
end tuple
```

```
fun estufa_voraz(T: Set of Troncos, K1: float, K2: float) ret res: List of Troncos
    var T_aux: Set of Troncos
    var t: Troncos
    var minutos_pasados: nat

    T_aux := copy_set(T)
```

```

res := empty_list()
minutos_pasados := 0

while minutos_pasados ≤ 840 do
  if minutos_pasados ≤ 480 then
    t := elegir_tronco(T_aux, K1)
  else
    t := elegir_tronco(T_aux, K2)
  fi
  addr(res, t)
  minutos_pasados := minutos_pasados + t.tiempo
  elim(T_aux, t)
od

destroy_set(T_aux)
end fun

fun elegir_tronco(T: Set of Troncos, K: float) ret res: Troncos
var T_aux: Set of Troncos
var t: Troncos
var maximo_tiempo_ardiendo: nat

T_aux := copy_set(T)
maximo_tiempo_ardiendo := -∞

while not is_empty_set(T_aux) do
  t := get(T_aux)
  if t.tiempo > maximo_tiempo_ardiendo ∧ t.temperatura ≥ K then
    maximo_tiempo_ardiendo := t.tiempo
    res := t
  fi
  elim(T_aux, t)
od

destroy_set(T_aux)
end fun

```