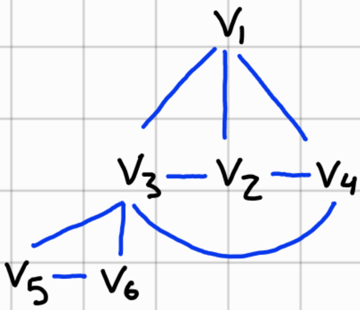
2) Un *modelo de intervalos* es una secuencia $\mathcal{I} = [s_1, t_1], \ldots, [s_n, t_n]$ de intervalos cerrados tales que $0 \leq s_1 \leq s_2 \leq \ldots \leq s_n$. El *grafo de intervalos* de $\mathcal{I}$ es el grafo $G(\mathcal{I})$ con n vertices $v_1, \ldots, v_n$ tal que v_i y v_j son adyacentes si y solo si $t_i \geq s_j$ para todo $1 \leq i < j \leq n$.

a) Proponer un algoritmo goloso que, dado un modelo de intervalos $\mathcal{I}$ cuyo grafo $G(\mathcal{I})$ es conexo, encuentre un árbol v_1-geodésico de $G(\mathcal{I})$. Recordar que un árbol T es v_1-geodésico cuando la distancia entre v_1 a v_i en T es igual a la distancia entre v_1 y v_i en G para todo $1 \leq i \leq n$. **Ayuda:** recordar el trabajo práctico, observando qué propiedad cumplen los intervalos correspondientes a cualquier camino de v_1 a v_i.

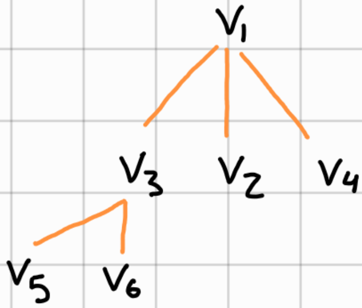
b) Demostrar que el algoritmo propuesto es correcto. **Ayuda:** recordar el trabajo práctico.

Complejidad: la complejidad temporal del algoritmo resultante debe ser $O(n)$.

v_1: [1, 4]
v_2: [2, 4]
v_3: [2, 8]
v_4: [4,6]
v_5: [7,10]
v_6: [8, 12]



$G(\mathcal{I})$



T v_1-geodésico

```
T ← ∅             T v1-geodésico representado por sus aristas.
i ← 1             El vértice padre del nivel de T que estamos construyendo.
j ← 2             El hijo potencial de i.
m ← 2             El hijo de i con máximo t es el siguiente i.
while j < n:
    if t_i ≥ s_j:
        T ← T ∪ {(i,j)}        Usamos la arista (i,j) en T.
        if t_j > t_m:
            m ← j              Actualizamos el siguiente i.
        j ← j + 1
    else:
        i ← m                  No hay más hijos para agregar, vamos al
                               siguiente nivel desde m.
```

Como G es conexo el ciclo siempre termina pues necesariamente cuando $t_i < s_j$, vale que $t_m \geq s_j$.

La elección greedy es utilizar a i como padre si existe arista entre i y j.