

Ejercicio 4 : Ordenación por inserción dicotómica

- **Precondición:**
 - t debe ser una lista de números enteros
 - Los elementos en t pueden ser ordenados según un criterio definido (en este caso de menor a mayor)
- **Poscondición:**
 - Al finalizar la ejecución del algoritmo, t debe estar ordenada de menor a mayor
 - Los elementos originales de t permanecen en la lista, pero reordenados.
- **Datos de entrada:**
 - Lista de numeros enteros
- **Datos de salida:**
 - Lista auxiliar de numeros enteros ordenada de menor a mayor
- **Tratamiento de datos:**
 - **Entrada de datos:** La lista inicial t contiene una serie de elementos (números en nuestro caso) que necesitan ser ordenados. Estos elementos pueden estar en cualquier orden y no se necesita ninguna preparación especial antes de aplicar el algoritmo.
 - **Procesamiento iterativo:** El algoritmo procesa cada elemento de la lista uno por uno, comenzando con el segundo elemento (ya que el primer elemento se considera trivialmente ordenado por sí solo).
 - **Búsqueda de la posición correcta:**

Para cada elemento seleccionado, se realiza una búsqueda binaria dentro de la porción de la lista que ya ha sido ordenada (los elementos antes del elemento actual). La búsqueda binaria divide repetidamente esta sublista en dos mitades y compara el elemento con el punto medio para determinar en qué mitad debería continuar la búsqueda.
 - **Inserción en lugar adecuado:**

Una vez encontrada la posición correcta mediante la búsqueda binaria, el elemento se inserta en esta posición. Esto puede requerir el desplazamiento de los elementos que ya están ordenados y que son mayores que el elemento a insertar, para hacer espacio para el nuevo elemento.
 - **Actualización de la lista:**

La lista t se actualiza en cada iteración con el nuevo elemento insertado en la posición correcta, asegurando que la porción procesada de la lista siempre permanezca ordenada.
 - **Salida de datos:**

Al finalizar todas las iteraciones, la lista t estará completamente ordenada y se devolverá como resultado final del algoritmo.

Ejercicio 5 : Ordenación topológica

- **Precondición:**
 - n debe ser un entero que representa el número de tareas.
 - restricciones debe ser una lista de pares de enteros donde cada par (i, j) indica que la tarea i debe ser completada antes de la tarea j .
- **Poscondición:**
 - Si es posible, se devuelve una lista de tareas ordenadas de acuerdo con las restricciones dadas.
 - Si no es posible (debido a ciclos en las restricciones), se devuelve None.
- **Datos de entrada:**
 - Número entero n que indica el número de tareas.
 - Lista de pares de enteros que representan las restricciones entre las tareas.
- **Datos de salida:**
 - Lista de tareas ordenadas de acuerdo a las restricciones, o None si no es posible realizar la ordenación.
- **Tratamiento de datos:**
 - **Entrada de datos:** Se recibe el número de tareas y las restricciones entre estas tareas. Las tareas están numeradas y las restricciones indican un orden que debe ser respetado.
 - **Construcción del grafo:**
 - Se crea una lista de sucesores para cada tarea, representando las tareas que dependen directamente de cada tarea.
 - Se calcula el grado de entrada para cada tarea, que representa cuántas tareas deben completarse antes de que esta pueda iniciar.
 - **Inicialización de la cola:**
 - Se inicializa una cola con todas las tareas que no tienen predecesores (grado de entrada es cero), es decir, tareas que pueden comenzar inmediatamente.
 - **Procesamiento iterativo:**
 - Mientras la cola no esté vacía, se extrae una tarea de la cola y se añade al orden topológico.
 - Se revisan todas las tareas que dependen de la tarea extraída, y se reduce su grado de entrada en uno.
 - Si alguna de estas tareas ahora no tiene predecesores, se añade a la cola.
 - **Verificación de ciclos:**
 - Si después de procesar todas las tareas, el orden topológico contiene todas las tareas, se devuelve este orden.
 - Si no contiene todas las tareas, significa que hay un ciclo y no es posible realizar una ordenación topológica, devolviendo None.
 - **Salida de datos:**

- Se devuelve el orden topológico de las tareas si es posible, o None si se detecta la presencia de ciclos y no se puede establecer un orden.