

Fase 1:

Método	Función Big-Oh	Explicación
union	$O(n)$	En este método existen dos bucles, pero no están anidados por lo tanto en el de los peor de los casos se completarán los dos.
getCampusCity	$O(n)$	En ambas opciones tendrá la misma complejidad, ya que solo se ejecutará una parte del "if", es decir un único bucle.
locateByCity	$O(n)$	El método consiste en un bucle que se recorre entero, por lo tanto, el peor de los casos es que la lista tenga n elementos.
orderBy	$O(n^2)$	Este método diferencia dos opciones y en ambas se ejecutan dos bucles, uno anidado en el otro, por lo tanto, una complejidad cuadrática
getStudentsByDataInterval	$O(n)$	Se trata de un bucle que recorre siempre la lista entera.

Fase 2:

Un árbol no es la estructura mas eficiente para almacenar datos secuenciales, ya que el siguiente siempre será mayor que el anterior. Nos quedaría una estructura tipo lista doblemente enlazada, relacionando cada dato con el posterior y el anterior, este tipo de estructura de datos sería el más eficiente para el almacenamiento y búsqueda de este tipo de datos.

Fase 3:

La mejor representación posible para un grafo con tantas aristas será una lista de vértices y otra lista compuesta de listas que almacenen los vértices adyacentes. Es la mejor ya que las listas pueden aumentar su tamaño sin dificultad y no almacenaran aristas que no existan, como ocurriría en una matriz.