

Comparativa de contenedores de la STL

Ejercicio optativo (Estructuras de Datos)

Objetivo

El objetivo de este ejercicio es que el estudiante realice un análisis experimental de las distintas estructuras de datos que ofrece la STL, que permita mostrar las diferencias de eficiencia empírica entre distintas estructuras para distintas operaciones.

El conocimiento de estas diferencias es fundamental para escoger en cada caso la mejor estructura de datos según las necesidades del sistema que se esté desarrollando.

El ejercicio

Se van a comparar los siguientes contenedores presentes en la STL:

- `vector<int>`
- `list<int>`
- `set<int>`
- `unordered_set<int>`

Crearemos un contenedor de cada tipo con los números del 1 al N , que será un parámetro del programa indicando el tamaño de la estructura.

Las operaciones que vamos a testear son las siguientes:

- Inserción de un elemento en la estructura: compararemos un `push_back` en el caso de `vector` y `list`, y un `insert` en el caso de `set` y `unordered_set`, pasando al método `insert` únicamente el elemento a insertar.
- Borrado del elemento central de la estructura: se borrará el elemento en la posición $\lfloor \frac{N}{2} \rfloor$ en el caso de `vector` y `list`, y el elemento con clave $\lfloor \frac{N}{2} \rfloor$ en el caso de `set` y `unordered_set`.
- Acceso al elemento central de la estructura: con las mismas consideraciones que en el apartado anterior.
- Búsqueda del elemento con clave/valor N en la estructura:
 - En el caso de `vector` se implementará una búsqueda binaria: consultando el elemento en la posición $\lfloor \frac{N}{2} \rfloor$ y, si no es el valor buscado, se busca en la mitad

superior o inferior del vector según si el valor encontrado es menor o mayor que el buscado.

- En el caso de `list`, la única forma de buscar un elemento es recorrer la lista, aunque esté ordenada, puesto que no contamos con acceso aleatorio.
- En el caso de `set` y `unordered_set` bastará con usar el método `find`.

Variando el tamaño del problema con sucesivas llamadas, mediremos el tiempo para cada operación sobre cada estructura, como ya hicimos en la práctica sobre eficiencia. A continuación generaremos una gráfica para cada operación, donde se comparen las eficiencias de cada contenedor. En cplusplus.com tenemos toda la información de todas estas estructuras de datos: en particular, para cada método de la clase, tenemos información sobre la complejidad de la operación, con lo que se comprobará si lo reflejado en las gráficas se corresponde con la información proporcionada.