

1. Reimplemente los ejercicios 1, 2, 3 y 4 de la práctica 11 para utilizar contenedores asociativos.
 - o Cómo haría la elección del más adecuado?
 - o Cambiaría su criterio si desea que se pueda acceder a los nombres de los productos comprados de forma ordenada (por ejemplo alfabética)?
2. Instancie contenedores `std::list<int>`, `std::deque<int>` y `std::vector<int>`. Agrégueles enteros con valor random. Ordénelos con `std::sort` midiendo el tiempo necesario para realizar tal operación. Repita el procedimiento cambiando la cantidad de elementos en los contenedores. Compruebe el orden del algoritmo.
3. Sobre uno de los contenedores del ejercicio 2, aplique la función template `for_each` que se encargue de duplicar cada uno de los valores contenidos.
4. Utilizando el contenedor resultado del ejercicio 3, aplique la función template `all_of` para verificar que todos los valores contenidos son pares.
5. Qué está haciendo el siguiente código:

```
for (auto i = start; i != end; ++i)
    std::rotate(std::upper_bound(start, i, *i), i, std::next(i));
```

6. Qué está haciendo la siguiente función:

```
template<class FwdIt, class Compare = std::less<>>
void xxxxx(FwdIt first, FwdIt last, Compare cmp = Compare{}) {
    auto const N = std::distance(first, last);
    if (N <= 1)
        return;
    auto const pivot = std::next(first, N / 2);
    std::nth_element(first, pivot, last, cmp);
    xxxxx(first, pivot, cmp);
    xxxxx(pivot, last, cmp);
}
```