

Homeworks

En la presente tarea encontrará un **Makefile**. Al ejecutar el comando

```
make test-example
```

podrá ver los resultados de ejecutar los tests especificados en `example-list/list-test.cc`. Al hacerlo, podrá ver que los tests *pasan*.

Los tests son ejecutados desde dentro de un contenedor Docker, de manera que no es necesario que instale google test en su computadora.

Similarmente, al ejecutar cualquiera de los siguientes comandos

```
make test-hw1
make test-hw2
make test-hw3
make test-hw4
make test-hw5
```

podrá ver los resultados de ejecutar los tests especificados en `hw1/hw1-test.cc`, `hw2/hw2-test.cc`, `hw3/hw3-test.cc`, etc.. Al hacerlo, notará que *estas tests no pasan*.

El homework 1 consiste en implementar las funciones listadas en `hw1/hw1.h`. El homework 2 consiste en implementar las funciones listadas en `hw2/hw2.h`. Y así para todos los homeworks.

Los test cases incluidos en `test-hw1.cc`, `test-hw2.cc`, etc. **NO ESTÁN COMPLETOS, AL MOMENTO DE LA CORRECCIÓN SE USARÁN MUCHOS MAS TEST CASES**. Todas las funciones listadas en los archivos `hw1/hw1.h`, `hw2/hw2.h`, `hw3/hw3.h`, etc. serán testeadas exhaustivamente en los test cases que se utilizarán. En los test cases que están provistos, solo algunas funciones se testean, y de manera superficial. Se alienta al alumno a desarrollar sus propios test cases, a ser capaz de pensar condiciones de borde y todos los posibles escenarios en los que es importante testear los algoritmos siendo implementados.

Los archivos `hw1/hw1.h`, `hw1/hw1_impl.h`, `hw2/hw2.h`, `hw2/hw2_impl.h`, etc. pueden modificarse a gusto del alumno. Solo que los prototipos de las funciones que se implementan se deben respetar. Deben ser tal como indicados en los archivos `hw1.h`, `hw2.h`, etc., porque de esa manera serán usados en los unit tests.

Como se mencionó, su código será testeado con un número adicional de tests que no son provistos. Nótese que el archivo de tests **NO** puede modificarse. Esto implica, entre otras cosas, que la forma en la que debe incluirse su trabajo en ese archivo es como se hace en este momento: a través de la línea `#include "hw1.h"`.

Si en cualquiera de los homeworks decide no utilizar la estructura habitual para implementaciones que usan templates (header-only, con header de implementación separados),

y opta por tener archivos `.cc` adicionales, puede agregarlos en la línea del `Makefile` que contiene la lista de archivos `.cc` a compilar. Esta es: usted puede modificar en el `Makefile` la línea:

```
CC_HW1=hw1/hw1-test.cc
```

para que diga, por ejemplo:

```
CC_HW1=hw1/hw1-test.cc hw1/archivo_nuevo_1.cc hw1/archivo_nuevo_2.cc
```

FECHAS DE ENTREGA

- Homework 1: ya entregada.
- Homework 2: 11 de noviembre a las 23:59.
- Homework 3: 18 de noviembre a las 23:59.
- Homework 4: 25 de noviembre a las 23:59
- Homework 5: 02 de diciembre a las 23:59.

Entregables en canvas

Deberá entregar un archivo `.zip` conteniendo los archivos y directorios en exactamente la misma estructura que como les fueron provistos, pero conteniendo también su versión de los archivos. Cuando entregue el homework2, no es necesario que en el zip que entregue estén los archivos correspondientes al homework3, homework4 y homework5. Similarmente, cuando entregue por ejemplo el homework4 no es necesario que entregue nuevamente los archivos correspondientes a homeworks previos.

A los archivos que decidió eliminar, simplemente elimínelos. En los archivos que realizó cambios, incluya su versión.