

CS2013 - Programación III
Práctica Calificada #3 (PC3)
2023 - 2

Profesor: Rubén Rivas

heap/hash - 10 puntos

Dado un **texto** de caracteres (`std::string` o `std::wstring`) y dado un valor **n** que sirve para remover **n** caracteres del texto de modo que al realizar la suma de los cuadrados de las cantidades de cada carácter se obtenga el valor mínimo posible.

Diseñar el template de función **minimum_sum_squared_count** que retorne la suma mínima de cuadrados del conteo de cada carácter.

Nota: se sugiere usar **hash** para contar los caracteres y **heap** para ordenar los **N** caracteres, asegurese de obtener una complejidad **O(n)**

Casos de uso #1

```
std::cout << minimum_sum_squared_count("aabaaacc"s, 3) << std::endl;
```

Casos de uso #2

```
std::wstring text = L"universidad de ingenieria y tecnologia";  
std::cout << minimum_sum_squared_count(text, 4);
```

Casos de uso #3

```
auto text = L"Universidad de Ingenieria y tecnologia"s;  
std::cout << minimum_sum_squared_count(text, 8);
```

Casos de uso #4

```
auto text = ""s;  
std::cout << minimum_sum_squared_count(text, 20);
```

hash - 10 puntos

Dada una matriz de $N \times N$, Inicialmente, con cada celda vacía. Y dada K actualizaciones de las celdas de la Matriz, donde cada actualización representada por (r,c) actualizará toda la fila r y toda la columna c .

Desarrollar una clase template (functor) **count_empty_cells** que permita calcular y retornar el número de celdas vacías en la matriz al finalizar las actualizaciones.

Tip: Utilizar 2 tablas hash, una para filas y otra para columnas

Caso de uso #1

```
count_empty_cells<20> counter;
counter.add(0,0);
counter.add(19,19);
std::cout << counter() << std::endl;
```

Caso de uso #2

```
count_empty_cells<15> counter;
counter.add(0,0);
counter.add(1,0);
counter.add(0,3);
counter.add(4,3);
counter.add(19,19);
std::cout << counter() << std::endl;
```

Caso de uso #3

```
count_empty_cells<15> counter{
    {0, 0},
    {1, 0},
    {0, 3},
    {4, 3},
    {11, 2},
    {5, 11},
    {19, 19},
};
std::cout << counter() << std::endl;
```

Caso de uso #4

```
std::vector<pair<size_t, size_t>> v = {
    {0, 0},
    {1, 0},
    {0, 3},
    {4, 3},
    {11, 2},
    {5, 11},
    {19, 19},
};
count_empty_cells<45, std::vector> counter(v);
std::cout << counter() << std::endl;
```