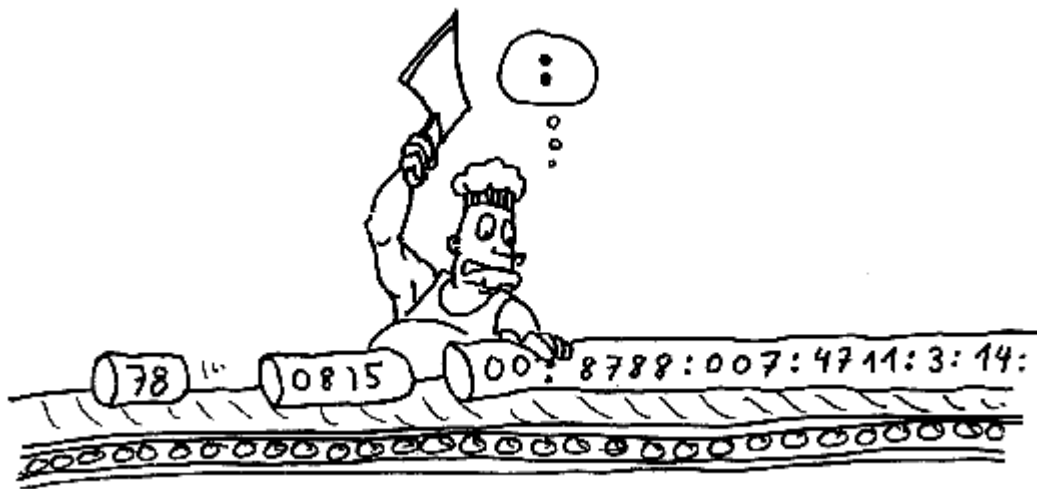


Práctica 1: Tokenizador



Cálculo de la complejidad temporal del método Tokenizar en casos especiales

Para calcular la complejidad de este método hay que hacer referencia a qué hay dentro de él, como se puso la captura anteriormente cuando se terminó de implementar las URLs. Como no he podido completar todos los casos especiales haré el cálculo de la complejidad únicamente sobre este caso (por desgracia :c).

Habiendo un **bucle principal** para recorrer todo el string será $O(n)$ y al usarlo de manera eficiente para recorrer las URLs pudiendo así **no** tener que **introducir otro bucle anidado** y **avanzar el índice una vez recorrida la URL** se queda con una **complejidad temporal $O(n)$** . Concluyendo como **caso peor $O(n)$** y como **caso mejor $\Omega(n)$** . En ambos casos, haya casosEspeciales o no los haya obtendremos $O(n)$.

Cálculo de la complejidad espacial del método Tokenizar en casos especiales

Para calcular la complejidad de este método hay que hacer referencia a qué hay dentro de él y cómo se inicializan las variables que se usan, como se puso la captura anteriormente cuando se terminó de implementar las URLs. Como no he podido completar todos los casos especiales haré el cálculo de la complejidad únicamente sobre este caso (por desgracia :c). Haciendo uso de la herramienta **memory** podemos saber cual es el **pico más alto de memoria** como ya se hizo en capturas anteriores. Viendo el contenido de este método podemos observar que al no tener que hacer el recorrido del string de delimitadores tenemos una eficiencia espacial muy grande también. Situándonos en su **caso mejor** sería **complejidad espacial $\Omega(1)$** ya que está hecho para que si tiene un input de todo delimitadores no guarde nada en memoria. La complejidad espacial en su **caso peor** será $O(n)$ ya que tendrá que almacenar en tokens y en palabras tokens de longitud n . Como se puede ver en esta imagen tenemos un impacto en memoria muy pequeño.

```
antonio@ubuntu:~/Escritorio/Universidad/4/ei/EI/Entrega1$ g++ main.cpp tokenizador.cpp -o main
antonio@ubuntu:~/Escritorio/Universidad/4/ei/EI/Entrega1$ ./memory ./main

Memoria total usada: 400 Kbytes
"   de datos: 268 Kbytes
"   de pila: 132 Kbytes
```