Tarea 1 - Búsqueda en texto

CC4102 - Diseño y Análisis de Algoritmos
Profesor: Gonzalo Navarro Auxiliar: Jorge Bahamonde
Ayudantes: Fabián Mosso - Sebastián Ferrada - Tomás Wolf

1 Introducción

En esta tarea deberá implementar y evaluar algoritmos de búsqueda en texto. El principal objetivo de esta tarea, sin embargo, es aplicar los conceptos y técnicas experimentales vistas en clases. Se espera que se implementen los algoritmos y se entregue un informe que indique claramente los siguientes puntos:

1. Las *hipótesis* escogidas antes de realizar los experimentos.
2. El *diseño experimental*, incluyendo los detalles de la implementación de los algoritmos, la generación de las instancias y las medidas de rendimiento utilizadas.
3. La *presentación de los resultados* en forma de una descripción textual, tablas y/o gráficos.
4. El *análisis e interpretación* de los resultados.

2 Los Algoritmos

Se le pide implementar los siguientes algoritmos:

- Fuerza Bruta
- Algoritmo Knuth-Morris-Pratt (KMP)
- Algoritmo Boyer-Moore (BM)

Para la descripción de cada algoritmo se considerará que el texto S tiene un largo n, y el patrón P a buscar tiene largo m.

Algoritmo de Fuerza Bruta

Una forma ingenua de realizar la búsqueda es alinear el patrón con el primer caracter del texto. A continuación, se compara cada caracter del patrón con el correspondiente del texto hasta llegar al final del patrón (en cuyo caso se anuncia que se ha encontrado una instancia del patrón) o encontrar una diferencia. En ambos casos, se desplaza el patrón una posición y se comienza la comparación de nuevo.

Algoritmo Knuth-Morris-Pratt (KMP)

La idea de KMP es modificar el algoritmo ingenuo de forma mover el patrón lo más posible, en vez de sólo una posición. Supongamos que se ha calzado $P[1, j]$ con $T[i - j + 1, i]$ y se encuentra una discrepancia ($P[j + 1] \neq T[i + 1]$). En este caso, el patrón puede ser desplazado de modo que el prefijo propio más largo de $P[1, j]$ que también sea sufijo de $P[1, j]$ esté alineado con el texto, con el último caracter de este prefijo alineado con $T[i]$. En otras palabras, si $f(j)$ es el número

tal que $P[1, f(j)]$ es el prefijo propio más largo que también es sufijo de $P[1, j]$, entonces el patrón puede desplazarse hasta que $P[1, f(j)]$ esté alineado con $T[i - f(j) + 1, i]$. A esta función $f(j)$ se le denomina *función de fracaso*:

$$f(j) = \max\{i < j \mid P[1, i] = P[j - i, j]\} \tag{1}$$

La función de fracaso se precalcula de la siguiente forma. Por definición, $f(1) = 0$. Al calcular $f(j + 1)$, se busca un $i + 1$ tal que el $i + 1$-ésimo caracter del patrón sea igual al $j + 1$-ésimo. Para ello, debe cumplirse que $i = f(j)$. Si $P[i + 1] = P[j + 1]$, entonces $f(j + 1) = i + 1$. Si no, se reemplaza i por $f(i)$ y se chequea nuevamente esta última condición.

Algoritmo Boyer-Moore-Horspool (BMH)

Este algoritmo recorre el patrón de forma inversa, de derecha a izquierda. Si se encuentra una discrepancia con el texto, se desliza el patrón de modo que el caracter del texto $T[k]$ que estaba alineado con $P[m]$ quede alineado con $P[j]$ (con $j < m$), si existe ese calce, o a la izquierda de $P[1]$. Llamamos $P[0]$ al caracter ficticio a la izquierda de $P[1]$.

Se define la función de salto s, que se precalcula, como:

- 0 si $T[k]$ no está en $P[1, m - 1]$
- j si la última aparición de $T[k]$ en $P[1, m - 1]$ es en la posición j.

Esta función se utiliza para deslizar el patrón tanto en el caso de una discrepancia como al encontrarse un calce completo: el patrón se desliza $m - s(c)$ caracteres, con c el caracter del texto que estaba alineado con $P[m]$.

3 Experimentos y Datos

Trabaje con los siguientes conjuntos de datos. Al final de este documento puede encontrar links que le pueden ser de utilidad para obtener datos reales:

Alfabeto binario

Genere un texto de al menos 1MB de caracteres binarios de forma aleatoria. Para cada largo en $\{2^2, ..., 2^7\}$, genere patrones aleatorios de este largo y busque todas sus ocurrencias en el texto.

ADN real

Escoja una subcadena de ADN de la fuente de datos indicada al final del documento de 1MB como texto. Es probable que deba tratar el texto antes de ejecutar los algoritmos sobre éste, removiendo los caracteres que no correspondan a las cuatro bases nitrogenadas (G, C, A, T). Asegúrese de que, después de este preprocesamiento, su texto siga teniendo al menos 1MB de caracteres. Para cada largo en $\{2^2, ..., 2^7\}$, obtenga patrones de este largo extraídos del mismo texto y busque todas sus ocurrencias.

ADN sintético

Genere un texto de al menos 1MB de caracteres, de forma uniforme, utilizando el alfabeto de la parte anterior ($\{G, C, A, T\}$). Para cada largo en $\{2^2, ..., 2^7\}$, obtenga patrones de este largo extraídos del mismo texto y busque todas sus ocurrencias.

Lenguaje natural, caso real

Escoja un texto de al menos 1MB de caracteres en algún idioma que considere adecuado. Convierta todos los separadores entre dos palabras en un único blanco (considere todo caracter fuera de a-z y A-Z como un separador).

Asegúrese de que, después de este preprocesamiento, su texto siga teniendo al menos 1MB de caracteres. Para cada largo en $\{2^2, ..., 2^7\}$, obtenga patrones extraídos de este largo del mismo texto y busque todas sus ocurrencias.

Lenguaje natural, caso sintético

Genere un texto de al menos 1MB de caracteres, de forma uniforme, utilizando el alfabeto de la parte anterior. Para cada largo en $\{2^2, ..., 2^7\}$, obtenga patrones extraídos de este largo del mismo texto y busque todas sus ocurrencias.

Mediciones

Mida, para cada algoritmo, cada conjunto de datos y cada largo de patrón:

- El tiempo de ejecución promedio para cada algoritmo y el error asociado a este promedio, así como la confianza sobre este error.
- El número promedio de comparaciones de caracteres realizadas por cada algoritmo y el error asociado a este promedio, así como la confianza sobre este error.

Además, para cada algoritmo, conjunto de datos y largo de patrón, repita los experimentos de modo de garantizar un 5% de error con un 95% de confianza Recuerde revisar las técnicas de experimentación para que su informe muestre su trabajo en una forma fácil de entender e interpretar. Además, explique claramente los pasos que siguió para sus experimentos, de modo de asegurar su reproducibilidad. Discuta apropiadamente los resultados que obtenga, considerando los resultados teóricos asintóticos para el mejor caso, peor caso y caso promedio para cada algoritmo.

4 Entrega de la Tarea

- La tarea puede realizarse en grupos de a lo más 2 personas.
- Se descontará un punto por día de atraso o fin de semana.
- Para la implementación puede utilizar `C`, `C++`, o `Java`. Para el informe se recomienda utilizar LaTeX.
- Escriba un informe claro y conciso. Las ponderaciones del informe y la implementación en su nota final son las mismas.
- La entrega será a través de U-Cursos y deberá incluir el informe junto con el código fuente de la implementación (y todas las indicaciones necesarias para su ejecución).

5 Links

- En `http://users.dcc.uchile.cl/~bebustos/apuntes/cc3001/BusqTexto/` puede encontrar explicaciones más gráficas de cómo funcionan los algoritmos.
- En `http://pizzachili.dcc.uchile.cl/texts.html` puede encontrar tanto datos de lenguaje natural como secuencias de ADN.