



Instituto Politécnico Nacional.

Unidad Profesional Interdisciplinaria de Ingeniería
campus Zacatecas.

Ingeniería en Sistemas Computacionales

Análisis de algoritmos.

Profesor: Roberto Oswaldo Cruz Leija.

Grupo: 3CM1

“RABIN KARP”

ALUMNA;

Alejandra Monserrath Esparza Ríos.

Zacatecas, Zac., a 14 de noviembre del 2019.

Introducción

Los algoritmos de búsqueda de subcadenas o en inglés string matching, son algoritmos donde se busca encontrar todas las ocurrencias de una subcadena denominada patrón dentro de un texto. Estos pueden ser simples o múltiples dependiendo de la cantidad de patrones que se busca.

Rabin Karp es un algoritmo de búsqueda de subcadenas simples, enunciado por Michael Oser Rabin y Richard Manning Karp en 1987.

Marco teórico.

Este algoritmo se basa en calcular la función de dispersión para la posición i del texto conociendo su valor para la posición $i-1$.

Se transforman los M caracteres en números agrupándolos en una palabra que podría tratarse como un número entero. Esto equivale a escribir los caracteres como números en un sistema base d , donde d es el número de caracteres posibles. El algoritmo Rabin-Karp calcula el valor hash para el patrón, y para cada subsecuencia de M -caracteres de texto. Si los valores hash son diferentes, se calcula un valor para la siguiente secuencia. Pero si son iguales se usa una comparación de Fuerza Bruta. Este algoritmo tiene una pequeña mejora en el algoritmo de fuerza bruta. La idea básica detrás de este algoritmo está en la utilización de una función hash. De ahí, si hay algún $h(d) = h(e)$ significa que la cadena d existe en el texto ' e '.

Cabe resaltar que este algoritmo se utiliza en la detección de plagio. Dado el material de origen, el algoritmo puede buscar rápidamente en un papel ejemplos de oraciones del material de origen, ignorando detalles como el caso y la puntuación. Debido a la abundancia de las cadenas buscadas, los algoritmos de búsqueda de una sola cadena no son prácticos aquí. De nuevo, el algoritmo Knuth-Morris-Pratt o el algoritmo de búsqueda de cadenas de Boyer-Moore es un algoritmo de búsqueda de cadenas de un solo patrón más rápido que Rabin-Karp. Sin embargo, es un algoritmo de elección para la búsqueda de múltiples patrones. Si queremos encontrar alguno de los números grandes, digamos k , patrones de longitud fija en un texto, podemos crear una variante simple del algoritmo de Rabin-Karp.

Para el texto de longitud N y P patrones de longitud combinada m , su promedio y mejor de los casos tiempo de ejecución es $O(n + m)$ en O espacio (p), pero su tiempo del peor caso es $O(nm)$.

Características de programación.

- ✚ Eficiente para patrones múltiples.
- ✚ Es relativamente rápido para la búsqueda de caracteres.
- ✚ Se basa en la utilización de hash.
- ✚ Desplazamiento: siempre uno a la derecha.
- ✚ Usa una función de Hashing (Rolling hash).

Pruebas de ejecución.

----- EMPATE DE CADENAS POR RABIN KARP-----

----- La cadena es:ostrfstfgustjyjystjystlogs

----- El patron a buscar es:st

El patron a buscar se ha encontrado en la posicion: 1

El patron a buscar se ha encontrado en la posicion: 5

El patron a buscar se ha encontrado en la posicion: 10

El patron a buscar se ha encontrado en la posicion: 16

El patron a buscar se ha encontrado en la posicion: 20

Se ha encontrado: 5 veces.

----- EMPATE DE CADENAS POR RABIN KARP-----

----- La cadena es:ostrfstfgustjyjystjystlogs

----- El patron a buscar es:y

El patron a buscar se ha encontrado en la posicion: 13

El patron a buscar se ha encontrado en la posicion: 15

El patron a buscar se ha encontrado en la posicion: 19

Se ha encontrado: 3 veces.

----- EMPATE DE CADENAS POR RABIN KARP-----

----- La cadena es:ostrfstfgustjyjystjystlogs

----- El patron a buscar es:z

Se ha encontrado: 0 veces.

----- EMPATE DE CADENAS POR RABIN KARP-----

----- La cadena es:sssseeeeeeeeeeddddddddddrrrrrrrrrrfgeeeeeee

----- El patron a buscar es:eee

El patron a buscar se ha encontrado en la posicion: 4

El patron a buscar se ha encontrado en la posicion: 5

El patron a buscar se ha encontrado en la posicion: 6

El patron a buscar se ha encontrado en la posicion: 7

El patron a buscar se ha encontrado en la posicion: 8

El patron a buscar se ha encontrado en la posicion: 9

El patron a buscar se ha encontrado en la posicion: 10

El patron a buscar se ha encontrado en la posicion: 11

El patron a buscar se ha encontrado en la posicion: 12

El patron a buscar se ha encontrado en la posicion: 13

El patron a buscar se ha encontrado en la posicion: 39

El patron a buscar se ha encontrado en la posicion: 40

El patron a buscar se ha encontrado en la posicion: 41

El patron a buscar se ha encontrado en la posicion: 42

El patron a buscar se ha encontrado en la posicion: 43

Se ha encontrado: 15 veces.

```
----- EMPATE DE CADENAS POR RABIN KARP-----  
----- La cadena es:sssseeeeeeeeeeddddddddtrrrrrrrrrfgeeeeeee  
----- El patron a buscar es:rrrrrr  
  
El patron a buscar se ha encontrado en la posicion: 28  
El patron a buscar se ha encontrado en la posicion: 29  
El patron a buscar se ha encontrado en la posicion: 30  
El patron a buscar se ha encontrado en la posicion: 31  
  
Se ha encontrado: 4 veces.
```

Conclusiones.

El algoritmo Rabin Karp calcula un hash para la comparación de las palabras, a cada unidad de comparación (letras, palabras, párrafos, etc) se le asigna una clave, de esta manera cuando se quiera comparar las unidades solo comparamos sus claves y no las letras, palabras, etc.

La desventaja de este algoritmo es que el sistema nunca puede saber qué documentos fueron los primeros. El algoritmo solo puede determinar que existen similitudes en los documentos comparables.