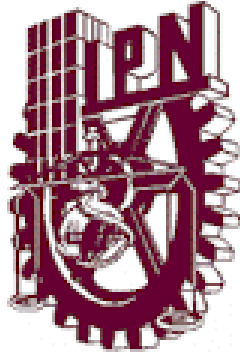


INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA CAMPUS
ZACATECAS



INGENIERÍA EN SISTEMAS COMPUTACIONALES

ANÁLISIS DE ALGORITMOS

Mtro. Roberto Oswaldo Cruz Leija

Empate de cadenas: Rabin – Karp

Daniela Margarita Honorato López

3CM1

14 de noviembre de 2019

Introducción

El algoritmo de Rabin-Karp es un algoritmo de búsqueda de cadenas creado por Richard M. Karp y Michael O. Rabin que utiliza el hash para encontrar cualquiera de un conjunto de cadenas de patrones en un texto.

En el algoritmo de Rabin-Karp, generaremos un hash de nuestro patrón que estamos buscando y verificaremos si el hash de nuestro texto coincide con el patrón o no. Si no coincide, podemos garantizar que el patrón no existe en el texto. Sin embargo, si coincide, el patrón puede estar presente en el texto.

```
1) AAAAAA AAAAAAAAAAAAAAAAAAAAAAAAH
   AAAAAH
   37≠100    1 compar
2) AAAAAA AAAAAAAAAAAAAAAAAAAAAAAAH
   AAAAAH
   37≠100    1 compar
3) AA AAAAAA AAAAAAAAAAAAAAAAAAAAAAAH
   AAAAAH
   37≠100    1 compar
4) AAA AAAAAA AAAAAAAAAAAAAAAAAAAAAAAH
   AAAAAH
   37≠100    1 compar
...
N) AAAAAAAAAAAAAAAAAAAAAAA AAAAAH
   AAAAAH
   5 compar          100=100
```

Valor Hash de “AAAAA” es 37

Valor Hash de “AAAAH” es 100

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 $O(p)$ espacio, pero su tiempo del peor caso es $O(nm)$.

Resultados

Caso 1:

Cadena ingresada:

OLAKEASE

Patrón a buscar:

KE

```
El patron se encuentra en la posicion: 3 de la cadena  
El patron ' KE ' se repite: 1 vez
```

Caso 2:

Cadena ingresada:

ANITALAVALATINA

Patrón a buscar:

ANITA

```
El patron se encuentra en la posicion: 0 de la cadena  
El patron ' ANITA ' se repite: 1 vez
```

Caso 3:

Cadena ingresada:

SOYUNPONYSALVAJE

Patrón a buscar:

H

```
El patron ingresado no se encuentra en la cadena ):
```

Conclusiones

El algoritmo Rabin-Karp a comparación del algoritmo de Fuerza Bruta calcula un valor hash para un patrón y para una subsecuencia de M caracteres. Si los valores hash obtenidos son iguales se emplea una comparación de Fuerza Bruta.

Hoy en día este algoritmo se sigue utilizando en algunos sistemas debido a que el consumo de memoria del Rabin-Karp permanece casi constante y crece lentamente al aumentar el número de patrones a buscar.

Bibliografía

S.e. (S.e). Introducción al algoritmo de Rabin-Karp. Noviembre 14, 2019, Sitio web: <https://riptutorial.com/es/algorithm/example/24653/introduccion-al-algoritmo-de-rabin-karp>

Cantos, P. (S.e). Algoritmia. Noviembre 14, 2019, Sitio web: http://bibing.us.es/proyectos/abreproy/12077/fichero/memoria%252Fpor_capitulos%252F02.algoritmia.pdf