PIMO - Grupo 1
Laboratorio Hashing
19 de abril de 2016 - Prof: Rodrigo López

Validando un diseño

Como parte del proceso de diseño de software, es frecuente escribir pequeños programas (llamados prototipos) que validan la viabilidad y la pertinencia de las decisiones de diseño. Una empresa que desarrolla juegos está planeando un juego en el que un super guerrero llega a una población desconocida y debe decidir si ataca solo o si cuenta con ayuda para atacar o, simplemente, no ataca. El criterio para tomar la decisión depende de la población (si es suficientemente pequeña, el guerrero atacará), en caso contrario si la religión practicada por la mayoría de los habitantes no es de origen judeo cristiano también atacará; en otro caso, no ataca. Para conocer esta información, en la plaza central de cada población hay un monumento con un dispositivo (que ha sido colocado por un espía) que emite las coordenadas geográficas del lugar.

El guerrero dispone de un celular con una tabla en donde se encuentra la información de todas las ciudades, la cual se accede (vía un mecanismo de hashing) a partir de las coordenadas geográficas. La información que interesa al guerrero es el nombre de la ciudad, población y religión practicada por la mayoría de sus habitantes. Sin embargo, también hay que almacenar las coordenadas para que el mecanismo de hashing funcione.

El prototipo

Con el prototipo se quiere probar una función de hashing adecuada para distintos tamaños de la tabla. Para ello se decide hacer lo siguiente.

1. Simular la inserción en la tabla de hashing a partir de un archivo de texto. En cada renglón del archivo aparecen —separados por coma— el nombre, latitud, longitud, número de habitantes y religión de cada población; por ejemplo

   ```
   Msida,35°53´33″N,14°28´58″E,5865,Budistas
   Ta´ Abram,36°03´54″N,14°14´00″E,8877,Catolicos
   ```

   Una coordenada se expresa en grados (dos dígitos y el símbolo °), minutos (dos dígitos y comilla) y segundos (dos dígitos y doble comilla), seguido de una letra (N,S,E,O).

   Para explicar el proceso de hashing que se quiere someter a prueba se utilizará la población `Msida` (primer renglón del ejemplo anterior). Recuerden que este proceso suele partirse en dos fases: hashcode y compress.

   hashcode:

   - Se consideran únicamente las coordenadas geográficas, en su orden: latitud y longitud; es decir 35°53´33″N y 14°28´58″E para producir un entero, formado por todos los dígitos de las dos coordenadas y la conversión a entero de la letra final de cada coordenada. Se ignoran los caracteres especiales. Para nuestro ejemplo,

     hashcode(35°53´33″N,14°28´58″E)=3553337814285879

compress:

- El entero obtenido por hashcode se convierte a un entero en el rango `[0..N-1]` para un N fijo. En este caso, se usará el método MAD (Multiply Add and Divide) con la siguiente fórmula:

$$compress(x, a, b, p, N) = ((ax + b) \ mod \ p) \ mod \ N$$

  En donde x es el resultado de hashcode.

Colisiones:

- Para esta parte del prototipo solamente se contabilizarán colisiones (ver más adelante)

2. Encontrar valores adecuados para los parámetros a, b, p, N de manera que se minimicen las colisiones. Para esto, los dos últimos renglones del archivo de datos son: un renglón en blanco, seguido de un renglón con una sucesión de enteros positivos separados por coma. Cada entero jugará el papel de N en la función compress. Ahora, para cada uno de estos valores se escogen valores (a, b, p) de la siguiente manera:

$$(N < p < 2N) \wedge (1 < a < p) \wedge (1 < b < p) \wedge (a \neq b) \wedge primo(p) \wedge primo(a) \wedge primo(b)$$

   Para cada combinación posible (a, b, p, N) se debe calcular el número total de colisiones que se producen al intentar armar la tabla de hashing de tamaño N.

3. Resultado. Para un archivo dado se debe imprimir:

   - Un primer renglón con el número de poblaciones procesadas.
   - A continuación, para cada valor de N —en el mismo orden de aparición de N en el archivo— se deben imprimir dos tuplas —separadas por un blanco— de la forma $(N, a, b, p, max)(N, a, b, p, min)$ con las combinaciones para las que se encontraron el máximo y el mínimo número de colisiones. Si hay diferentes combinaciones para las cuales se produce el máximo o el mínimo, se imprime la mayor de todas utilizando el siguiente orden:

$$(a, b, p) > (a', b', p') \equiv a > a' \vee (a = a' \wedge b > b') \vee (a = a' \wedge b = b' \wedge p > p')$$

   Por ejemplo, para un archivo con 200 poblaciones y cuyo último renglón es

```
109,127
```

   se imprime algo como:

```
200
(109,41,73,163,7) (109,41,13,151,0)
(127,83,7,149,5) (127,181,109,227,0)
```

Condiciones de Entrega

Subir a Moodle un archivo con las convenciones de nombre acostumbradas y terminando con el sufijo `_hash.py`. Por ejemplo, `lopezr_hash.py`