

RETO 5

Daniel Monjas Miguélez

19 de diciembre de 2019

1. Aparto a)

Como la función de rehashing doble tiene la forma de

$$h_i(k) = [h(k) + d_i] \% K \quad (1)$$

, se observa a primera vista que como el objetivo es que se puedan acceder a todos los posibles huecos de la tabla y teniendo en cuenta que la tabla tendrá K entradas debemos garantizar que para d_i sea capaz de tomar todos los elementos comprendidos en $I = 0, \dots, M - 1$, donde M es un número primo cualquiera. Para ello se adjunta un programa **di.cpp** que calcula los pares a,c que permiten que la función d_i tome todos los valores en el conjunto I, donde d_i está definida como,

$$d_i = [a * d_{i-1} + C] \% M \quad (2)$$

, donde $d_0 = 0$. Cualquiera de los pares que da como resultado el programa **di.cpp**, da lugar a un esquema de hashing válido.

2. Apartado b)

Como hemos visto arriba, la fórmula que se utiliza para calcular d_i hace el módulo M de $[a * d_{i-1} + C]$, luego no tiene sentido utilizar ni a, ni c mayores o iguales que M pues al final de la expresión se aplicará el módulo M. Tampoco tiene sentido utilizar a=0 e c=0 pues en ese caso la función d_i sería constante $\forall i \in I$, luego sabemos que

$$0 <= a < M \quad (3)$$

$$0 <= c < M \quad (4)$$

$$a \neq 0 \&\& b \neq 0 \quad (5)$$

, sabiendo esto y fijandonos en los resultados que hemos obtenido del programa **di.cpp**, tenemos que ver que **a** debe ser igual a 1 y **b** puede ser cualquier número entre 0 y M-1. Esto se debe a que si establecemos a=1 nos desplazamos en múltiplos de valores de c llegando así todos los elementos entre 0 y M-1.

*La única excepción es cuando M=2 donde a=0 y b=1 es también una solución.