

CC4102 - Diseño y Análisis de Algoritmos

Auxiliar 7

Prof. Gonzalo Navarro; Aux. Mauricio Quezada

16 de octubre de 2012

1 Bitmaps y `selectNext` (P2 C2 2011/2)

Considere el problema de `rank`. Este corresponde a contar la cantidad de unos hasta una posición i sobre una tira de bits B , `rank(B,i)`. Otra operación de interés sobre cadenas binarias es `selectNext(B,i)` que retorna la posición del siguiente 1 después de la posición i . Usando una metodología similar a la del problema de `rank` (uso de bloques y superbloques) diseñe una estructura que requiera $o(n)$ bits extra y que permita responder `selectNext(B,i)` en tiempo constante. Este problema es similar al de encontrar el sucesor al elemento i .

2 El problema de los k servidores

Considere el escenario donde tiene k puntos (*servidores*) en un *espacio métrico* (donde está definida una función de distancia d : simétrica, no-negativa y que cumple la desigualdad triangular) y una secuencia de puntos (*peticiones*) que debe atender. Cada vez que llega una petición, un servidor debe moverse hacia esa posición.

El problema *online* consiste en minimizar la distancia recorrida por todos los servidores luego de n peticiones, sin saber la secuencia de puntos a atender.

Recuerde que un algoritmo *online* ALG es r -competitivo si existe una constante a tal que para cualquier instancia I y el algoritmo óptimo OPT ,

$$\text{cost}_{ALG}(I) \leq r \cdot \text{cost}_{OPT}(I) + a$$

Donde r es el *radio competitivo*.

1. Sea ALG un algoritmo online para el problema de los k servidores bajo un espacio métrico arbitrario con al menos $k + 1$ puntos. Pruebe que el radio competitivo de ALG es al menos k .
2. Considere el problema en un *espacio métrico uniforme*, es decir, donde la distancia entre cada par de puntos distintos es 1. El algoritmo LRU (*least recently used*) es de la siguiente forma: si la petición ya está cubierta no hace nada, si no, moverá el servidor menos usado hacia la petición. Pruebe que LRU es k -competitivo.