EDA

13/09/2021

Dades professor:	
Nom: Gabriel	Cognom/s: Valiente
Mail: gabriel.valiente@upc.edu	Despatx: 234

1. Analitzar algoritmes

Per tal d'analitzar les funcions en els seus límits ho fem mitjançant notació asimptòtica.

$f \in O(g)$	$f \leq g$	$f \in o(g)$	$f < g$
$f \in \Omega(g)$	$f \geq g$	$f \in \omega(g)$	$f > g$
$f \in \Theta(g)$	$f = g$		

Notació O:

Si el càlcul del límit dona 0 vol dir que és O. $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$

Si el càlcul del límit dona 0 vol dir que és o. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$

Notació **Θ**:

Si el càlcul del límit dona $\mathbb{R}^+$ vol dir que és **Θ**. $\lim_{n\to\infty} \frac{f(n)}{g(n)} \in \mathbb{R}^+$

Notació Ω:

Si el càlcul del límit dona ∞ vol dir que és Ω. $\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$

Si el calcul del límit dona ∞ vol dir que és **ω**. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty.$

Ordre de creixement

Order of growth	Name	
$O(1) = O(n^0)$	Constant	Polynomial
$O(\log n)$	Logarithmic	Polynomial
$O(\sqrt{n}) = O(n^{1/2})$		Polynomial
$O(n)$	Linear	Polynomial
$O(n \log n)$	Quasi-linear	Polynomial
$O(n^2)$	Quadratic	Polynomial
$O(n^3)$	Cubic	Polynomial
$O(2^n)$		Exponential
$O(3^n)$		Exponential
$O(n!)$		Exponential

EDA

20/09/2021

Recurrències

Es un tipus de càlcul que ens permet calcular la eficiència dels algoritmes recursius. A diferencia dels iteratius compta les crides recursives. Les recurrències amb les que treballarem son d'una sola variable, així que si tenen mes d'una les haurem de combinar en nomes una.

Per resoldre una recurrència executem la seva definició reiteradament fins que arribem a una solució ja sigui exacta o asimptòtica.

El teorema de master

El teorema de master es pot aplicar si la part lineal es polinòmicament mes gran, mes petita o igual que la part recursiva.

El teorema de master es pot aplicar en el cas que la part lineal sigui major/igual a la recursiva però sense arribar al nivell de ser polinòmicament major.

$$T(n) = aT(n/b) + n^k$$

1. *If* $k < \log_b a$, *then* $T(n) = \Theta(n^{\log_b a})$.
2. *If* $k = \log_b a$, *then* $T(n) = \Theta(n^k \lg n)$.
3. *If* $k > \log_b a$, *then* $T(n) = \Theta(n^k)$.

2. Dividir i vèncer

Passos bàsics:

Dividir el problema en un nombre de subporoblemes que són instàncies mes petites del mateix problema

Vèncer els subproblemes solucionant-los recursivament

Cas base si el problema es suficientment petit soluciona'l directament

Combinar les solucions dels subproblemes per a solucionar el problema original

2.1 Mergesort

Per a ordenar un vector:

1. Es divideix el vector per la meitat en dos subvectors
2. Es venç ordenant recursivament els dos subvectors
3. Es combina els dos subvectors ordenats en un vector ordenat sencer

2.2 Quicksort

1. Es **divideix** el vector en dos vectors possiblement buits per un element q tal que el primer element del primer vector es menor o igual al element en la posició q i aquest segon element es menor igual a tots els elements del segon vector
2. Es **venç** ordenant els dos subvectors amb crides recursives a la funció
3. No es **combina** ja que estan ja ordenats en el seu lloc

2.3 Karatsuba

MULT(x, y, n)
    if $n = 1$ **then**
        return $x\,y$
    else
        partition x into a, b such that $x = a\,2^{n/2} + b$
        partition y into c, d such that $y = c\,2^{n/2} + d$
        $ac = \text{MULT}(a, c, n/2)$
        $bd = \text{MULT}(b, d, n/2)$
        $abcd = \text{MULT}(a + b, c + d, n/2)$
        return $ac\,2^n + (abcd - ac - bd)\,2^{n/2} + bd$

2.4 Strassen

MULT(A, B, n)
    let C be a new $n \times n$ matrix
    if $n = 1$ **then**
        $c_{11} = a_{11} \cdot b_{11}$
    else
        Partition A, B, and C into four $n/2 \times n/2$ matrices
        $S_1 = B_{12} - B_{22}$
        ...
        $P_1 = \text{MULT}(A_{11}, S_1, n/2)$
        ...
        $C_{11} = P_5 + P_4 - P_2 + P_6$
        ...
    return C

2.5 Best case Quicksort

BEST-CASE-QUICKSORT(A, p, r)
    if $p < r$ **then**
        $i = \lfloor (r - p + 1)/2 \rfloor$
        $x = \text{SELECT}(A, p, r, i)$
        $q = \text{PARTITION}(A, p, r, x)$
        BEST-CASE-QUICKSORT$(A, p, q - 1)$
        BEST-CASE-QUICKSORT$(A, q + 1, r)$