



Universitat de Lleida
Grau en Tècniques d'Interacció Digital i de Computació
Estructura de Dades

Pràctica 1

The Maximum Contiguous Subsequence Sum Problem (MCS)

Cristian Oprea i Xavier Vila

01 d'octubre de 2020

1 Algorisme d'ordre quadrat $O(N^2)$

Per arribar a aquesta solució vam eliminar el bucle *for* que controlava la variable *k*, la qual era totalment prescindible ja que no aportava res nou al codi, a part d'augmentar el seu cost.

2 Algorisme d'ordre lineal $O(N)$

Però per molt que haguéssim eliminat un bucle *for* el codi seguia sent terriblement ineficient degut a l'algorisme emprat, un *brute force algorithm*. Al principi no ens vam adonar d'això, així que vam seguir treballant fins aconseguir substituir el primer bucle *for* per un parell d' *ifs* (veure *maxSubsequenceSum_lineal_1* a *practica1.java*)

Després de fer una miqueta de recerca vam acabar concloent que l'algorisme de *Kadane* seria molt més eficient, ja que no caldria buscar totes les possibles arrays, només anar buscant la subarray més gran que acabi a l'índex *X*. Aquesta implementació (*maxSubsequenceSum_lineal_2* a *practica1.java*) s'aproxima bastant a la final, només que en aquest cas no funcionava amb els nombres negatius.

Però la gran pregunta segueix sent: Com funciona l'algorisme de *Kadane*?" doncs és molt senzill, va buscant la subarray més gran que acabi a cada índex (l'array fins a l'índex anterior + aquell índex) i la descarta si és negativa, ja que acabaria disminuint la suma final. Per fer que funcionés amb nombres negatius vam eliminar aquella comprovació i senzillament ens vam assegurar de que només suméssim en cas de que fes augmentar el valor màxim, si no resetejavem *thisSum*. Finalment vam aconseguir la funció *maxSubsequenceSum_lineal* en la qual vam aplicar tot el comentat i vam aconseguir la màxima eficàcia al mínim cost.