

Ejercicios sobre Algoritmos Voraces

Aitor Acedo Legarre

28 de Octubre de 2003

Powered by L^AT_EX.

1 Introducción del problema

El problema que tenemos entre manos consiste en diseñar un algoritmo que deberá encontrar el punto maximal de un conjunto de puntos. Para comprender el concepto de punto marginal de un conjunto de puntos tendremos que tener presente otro concepto como es el de punto dominante, se dice que un punto domina a otro cuando la coordenada x del primer punto es mayor o igual que la coordenada x del segundo punto y lo mismo con la segunda coordenada, siendo además puntos diferentes del espacio euclídeo. Para modelar el problema hemos utilizado una estructura para representar a un punto, la cual tendrá dos campos que serán dos enteros que representarán las coordenadas del punto. Para representar el conjunto utilizaremos un vector de dimensión fija, cuyo tamaño vendrá determinado por una constante. El algoritmo para el encontrar el maximal del conjunto de puntos simplemente asignará el primer punto como posible candidato para ser el maximal e intentará encontrar un punto de entre los puntos restantes un punto que lo domine, si lo encuentra este nuevo punto pasará a ser la solución y continuaremos recorriendo el vector hasta que lleguemos al final.

2 Algoritmo para resolver el problema

El algoritmo que nos va a resolver el problema tiene la siguiente estructura y está escrito en una nomenclatura similar a C:

```
#define TAM 100
struct point{ int x; int y};
struct point sol;
struct point C[TAM];

struct point *Maximal(struct point *C){
    int i;
    for(i=1; i<TAM; i++){
        if(C[i].x >= sol.x) && (C[i].y >= sol.y){
            if(C[i].x != sol.x) && (C[i].y != sol.y) {
                sol.x = C[i].x;
                sol.y = C[i].y;
            }
        }
    }
    return &sol;
}
```