TP1 - Algoritmos y Estructuras de Datos III

Catalina Gonzalo Juarros

2017-08-23

Índice

1. Descripción del problema

2. Resolución

Figura 1: Diego Peretti.

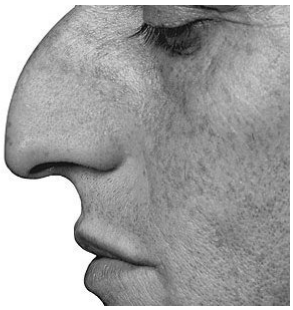

Figura 2: La nariz de Diego Peretti.

3. Complejidad

3.1. Caracterización del peor caso

El algoritmo, como vimos en la sección 2, consiste en probar subconjuntos de agentes hasta encontrar la máxima cantidad de informantes que pueden agregarse a la solución sin que uno contradiga a otro. Como es requisito que el arreglo que representa a cada subconjunto esté ordenado, sólo vamos a probar con **una** representación de cada subconjunto, por lo que la cantidad de soluciones posibles se corresponde con la cantidad de subconjuntos distintos de $\{1, ..., i\}$ (es decir, el *cardinal del conjunto de partes* de $\{1, ..., i\}$). Este número es 2^i. La justificación la voy a escribir cuando aprenda a hacer footnotes.
En el peor caso, el algoritmo tiene que probar **todos** los subconjuntos, o sea 2^i soluciones candidatas. Lo voy a justificar cuando efectivamente haya hecho el algoritmo.

3.2. Cálculo de complejidad

La complejidad de este algoritmo, en el peor caso, es

$$T(n) \in \mathcal{O}(2^i \times i^2 \times \log i \times a)$$

Justificación Dado que el algoritmo debe probar

4. Código fuente

5. Experimentación