

Presentación práctica de eficiencia

Asignatura: Algorítmica

Rubén Morales Pérez Francisco Javier Morales Piqueras
Bruno Santindrian Manzanedo Ignacio de Loyola Barragan
Lozano Francisco Leopoldo Gallego Salido

1 de mayo de 2016

Índice

Minimizando el número de visitas al proveedor

- Problema

- Diseño del algoritmo

- Demostración de la optimalidad

- Resultados empíricos

Minimizando el número de visitas al proveedor

Ordenador usado para la ejecución

Asus N56VJ

Sistema operativo: Linux mint (Rosa)

Memoria: 8GB

Procesador: Inter Core i7-4710HQ x 8

Gráficos: Nvidia geforce 750M

Tipo de SO: 64 bits

Disco: 1TB

Problema

Enunciado

Un granjero necesita disponer siempre de un determinado fertilizante. La cantidad máxima que puede almacenar la consume en r días, y antes de que eso ocurra necesita acudir a una tienda del pueblo para abastecerse. El problema es que dicha tienda tiene un horario de apertura muy irregular (solo abre determinados días). El granjero conoce los días en que abre la tienda, y desea minimizar el número de desplazamientos al pueblo para abastecerse.

Ejercicios

1. Diseñar un algoritmo greedy que determine en qué días debe acudir al pueblo a comprar fertilizante durante un periodo de tiempo determinado (por ejemplo durante el siguiente mes).
2. Demostrar que el algoritmo encuentra siempre la solución óptima.

Diseño del algoritmo

Algoritmo

Aplicamos greedy escogiendo en cada iteración el día más lejano que cumpla estas dos condiciones.

- ▶ Que diste menos de r días de la ultima reposición.
- ▶ Que la tienda esté abierta.

Representación

- ▶ Un vector de booleanos representa los días que abre la tienda.
Por ejemplo $[1, 0, 0, 1, 0, 1]$ significa que la tienda abre los días 1, 4 y 6.
- ▶ Una lista de enteros almacena los días que el granjero tiene que ir a comprar.

Ejemplo

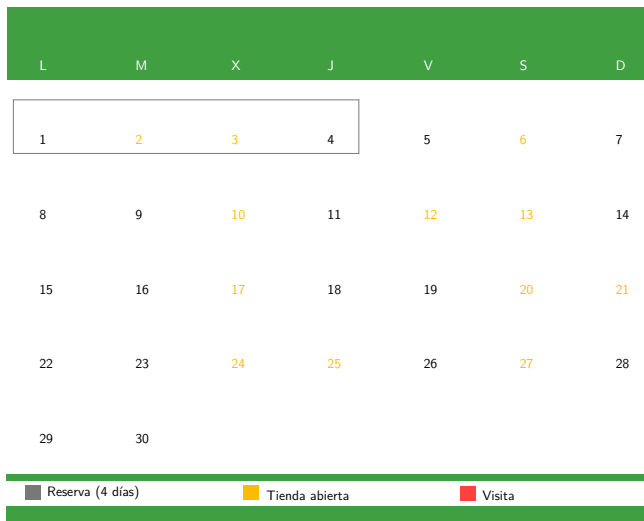
L	M	X	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Reserva (4 días)

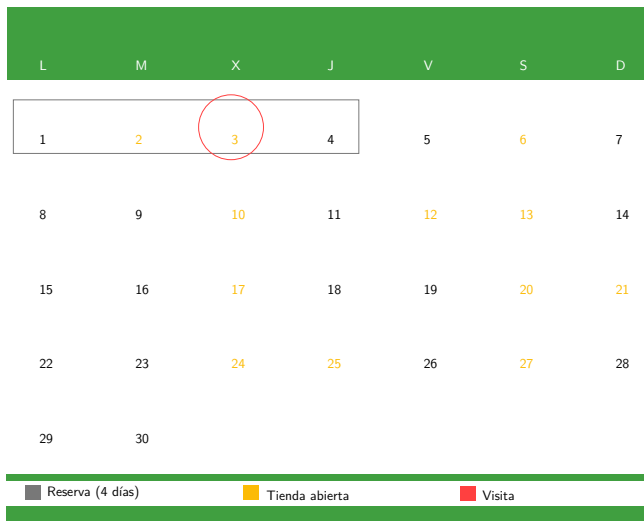
Tienda abierta

Visita

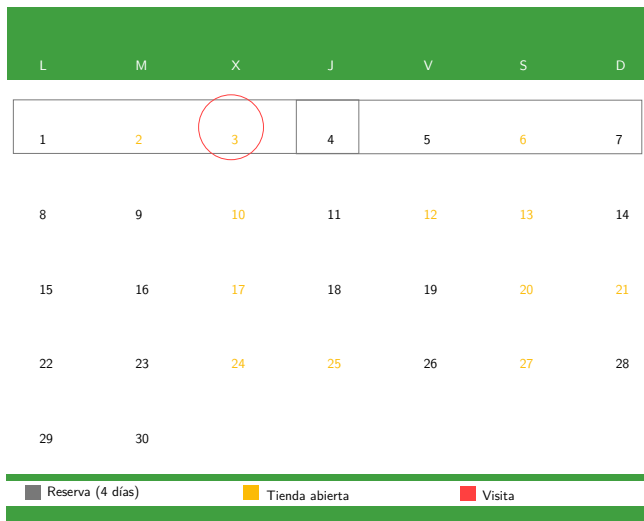
Ejemplo



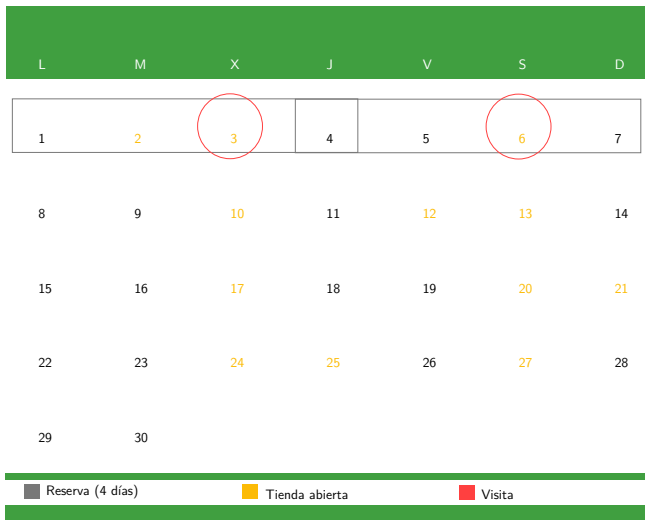
Ejemplo



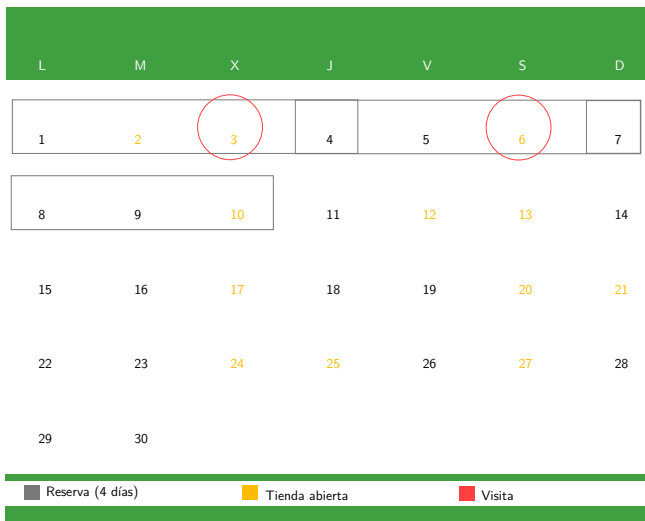
Ejemplo



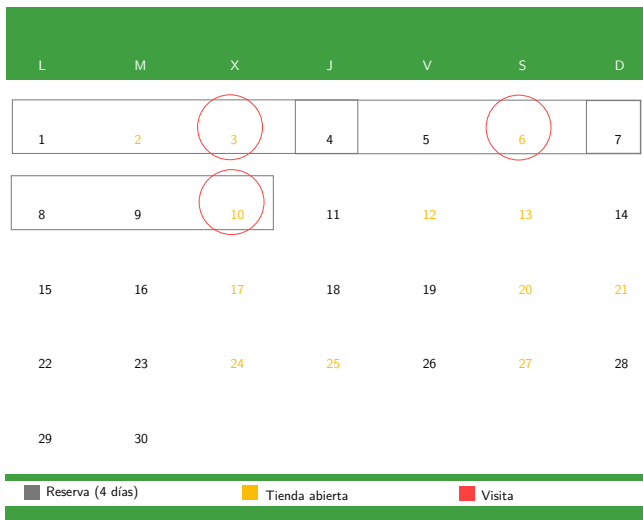
Ejemplo



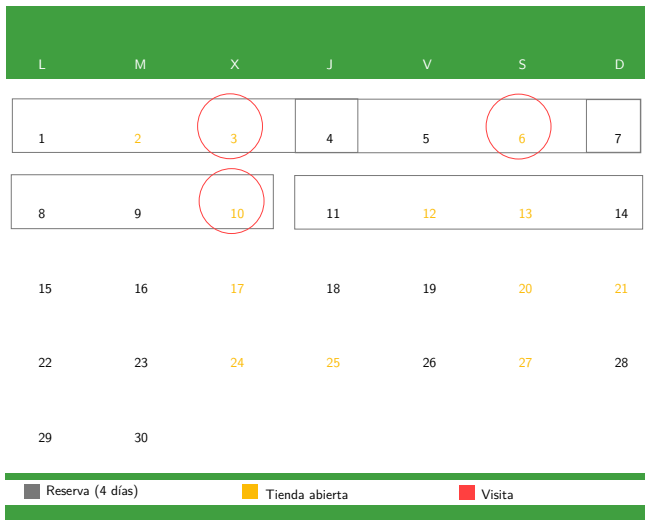
Ejemplo



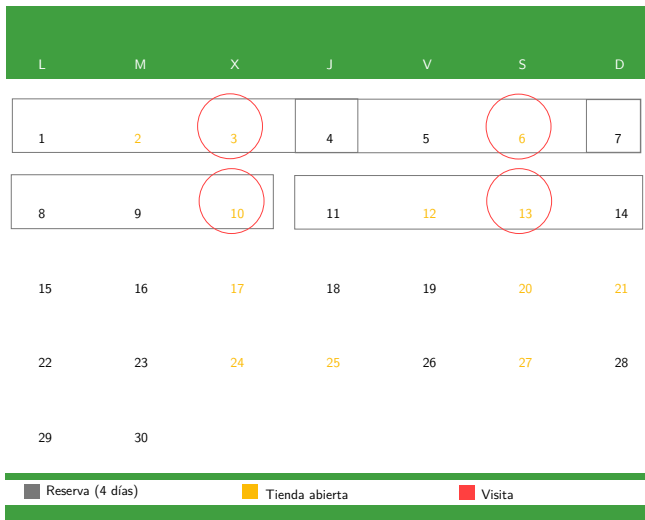
Ejemplo



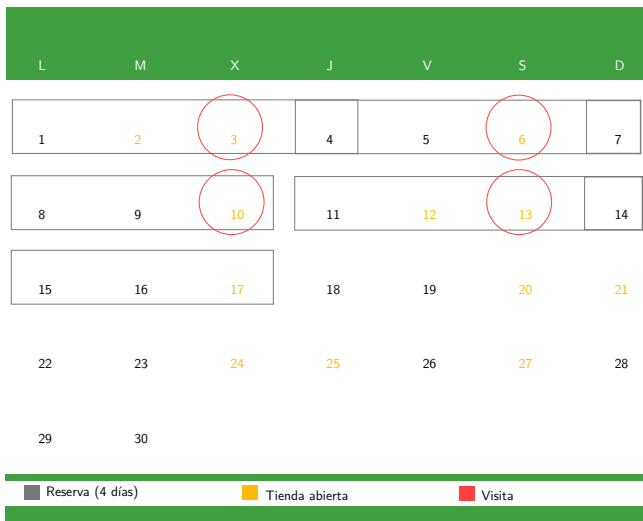
Ejemplo



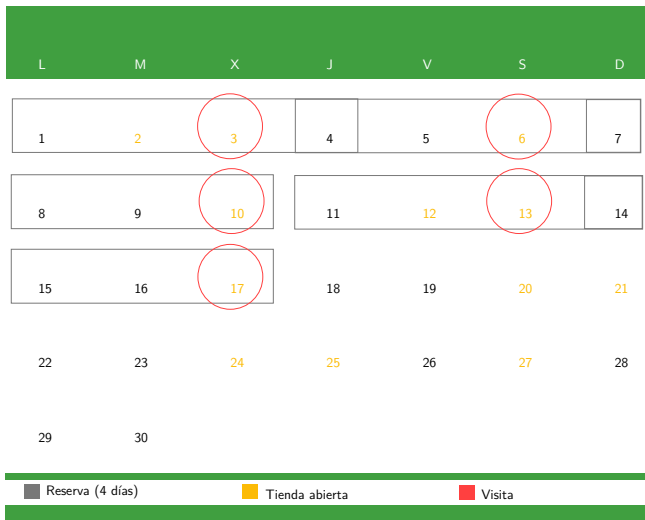
Ejemplo



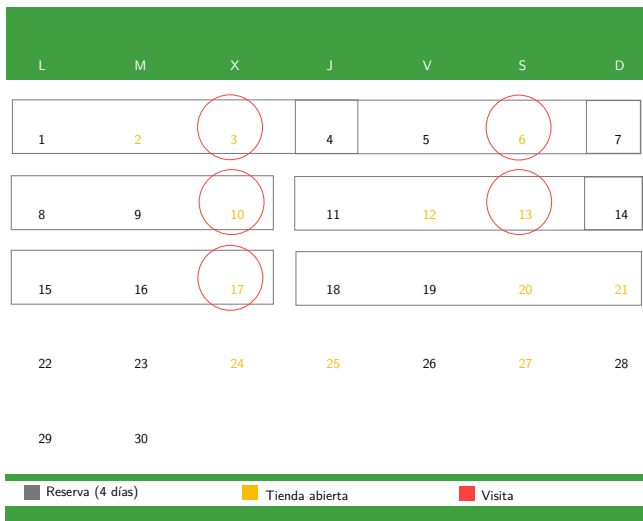
Ejemplo



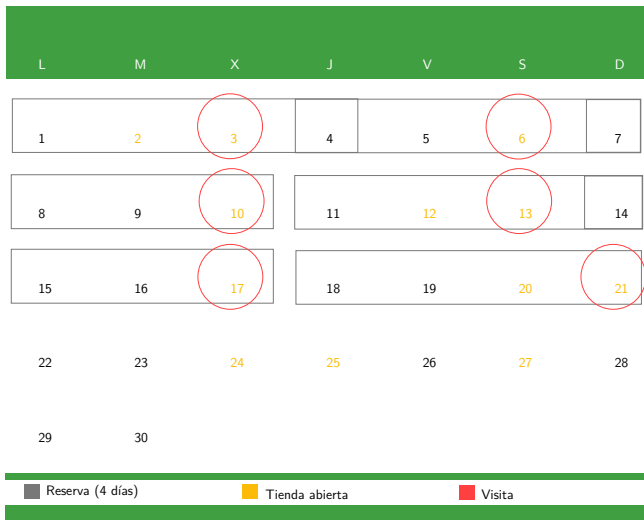
Ejemplo



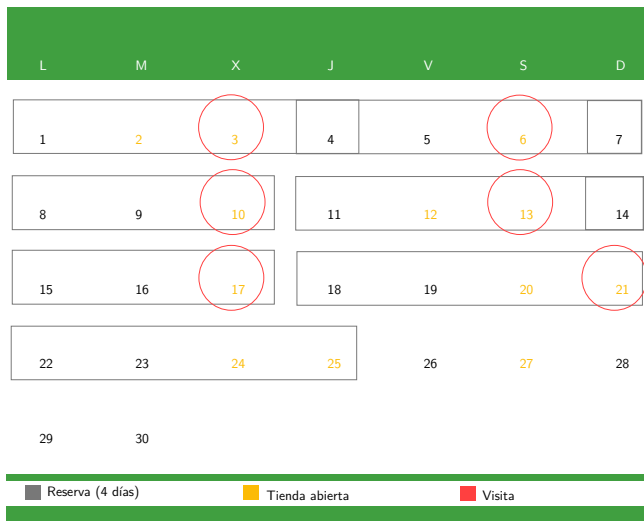
Ejemplo



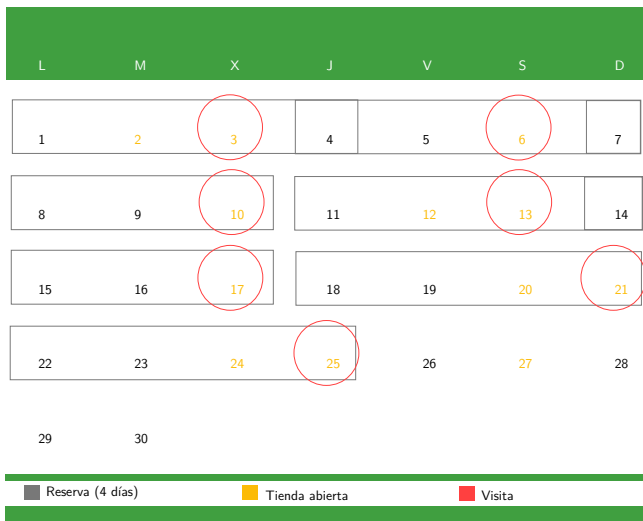
Ejemplo



Ejemplo



Ejemplo



Ejemplo



Ejemplo



Elementos de la demostración

- ▶ **p** la duración del periodo (ej: 30 días)
- ▶ **r** la duración de la reserva
- ▶ **a** vector de tamaño p que representa los días que abre la tienda

La tienda abre el día n si $a_n = 1$

La tienda no abre el día n si $a_n = 0$

- ▶ **d** el vector que almacena los días de visita
 $d_{i+1} = d_i + k_i / k_i \in \{x \in \mathbb{N} : 0 < x < r \text{ y } a_{d_i+x} = 1\}$

Demostración

Formulamos dos ecuaciones:

$$d_{i+1}^h = d_i^h + h_i \text{ donde } h_i = \max\{x \in \mathbb{N} : 0 < x < r \text{ y } a_{d_i+x} = 1\}$$

Podemos definir $n \in \mathbb{N} / d_n^h = \min\{d_i^h : d_i^h + r > p\}$ como el índice del último día que necesitamos ir a comprar.

$$d_{i+1}^t = d_i^t + t_i \text{ donde } t_i \in \{x \in \mathbb{N} : 0 < x < r \text{ y } a_{d_i+x} = 1\} / t_i < h_i \text{ para algún } i < n$$

Por las definiciones anteriores sabemos que

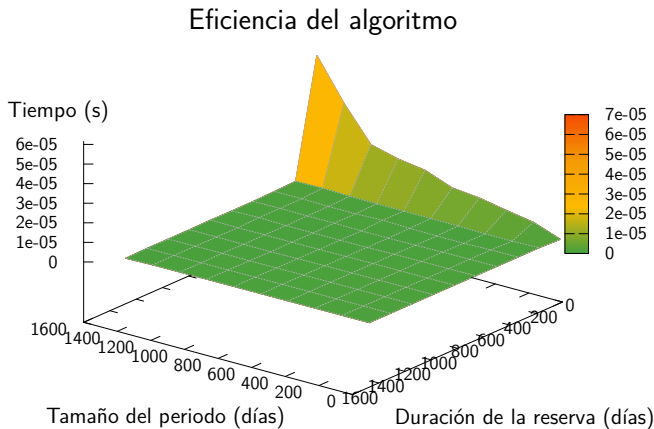
$$\exists m \in \mathbb{N} / d_m^t < d_i^h \forall i, m < i < n \Rightarrow d_n^t < d_n^h$$

Distinguimos dos casos:

1. Si $p - r < d_n^t < d_n^h$ las dos opciones son igual de buenas
2. Si $d_n^t < p - r < d_n^h$ la opción greedy es mejor

De aquí deducimos que utilizar el enfoque **greedy** en este caso siempre nos da el mejor resultado.

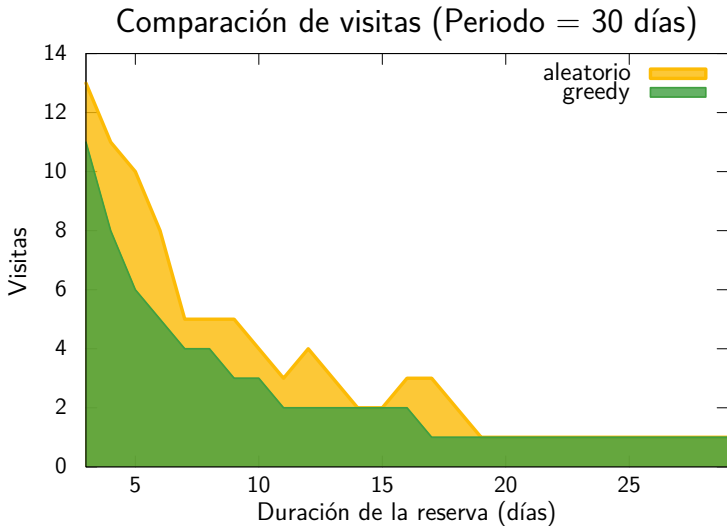
Eficiencia



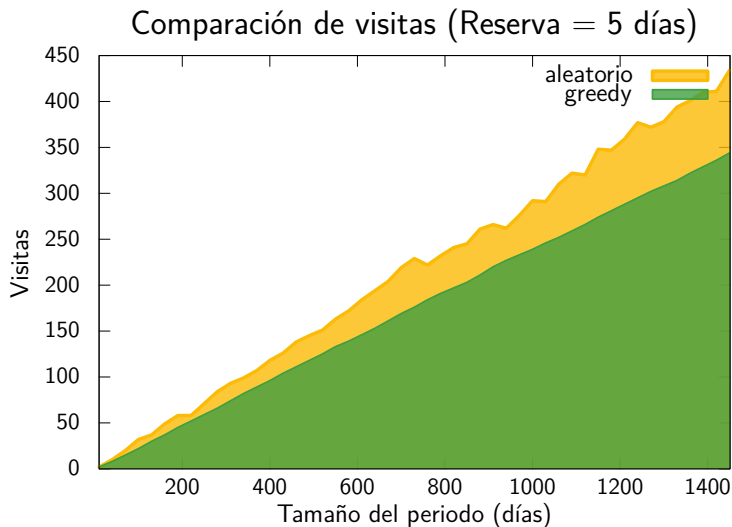
La eficiencia es $O(p)$.

En el caso de que el día abierto se encuentre en $\frac{r}{2}$ es $p - \frac{r}{2}$

Visitas fijando el periodo



Visitas fijando la reserva



Media de visitas (r aleatorio)

