



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Diseño de Algoritmos

Aplicación de técnicas

Algoritmos y Estructuras de Datos III  
Primer Cuatrimestre de 2015

| Integrante                  | LU     | Correo electrónico       |
|-----------------------------|--------|--------------------------|
| Baraňao, Facundo            | 480/11 | facundo_732@hotmail.com  |
| Confalonieri, Gisela Belén  | 511/11 | gise_5291@yahoo.com.ar   |
| Mignanelli, Alejandro Rubén | 609/11 | minga_titere@hotmail.com |
| Soliz, Carlos               | 406/12 | rcarlos.cs@gmail.com     |



Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

# Índice

|   |          |
|---|----------|
| <b>1. Objetivos generales</b>                         | <b>3</b> |
| <b>2. Plataforma de pruebas</b>                       | <b>3</b> |
| <b>3. Problema 1: Dakkar</b>                          | <b>4</b> |
| 3.1. Descripción del problema. . . . .                | 4        |
| 3.2. Desarrollo de la idea. . . . .                   | 4        |
| 3.3. Demostración de correctitud. . . . .             | 4        |
| 3.4. Análisis de complejidad. . . . .                 | 4        |
| 3.5. Experimentación y gráficos. . . . .              | 4        |
| 3.5.1. Test 1 . . . . .                               | 4        |
| 3.5.2. Test 2 . . . . .                               | 4        |
| <b>4. Problema 2: Zombieland II</b>                   | <b>5</b> |
| 4.1. Descripción del problema. . . . .                | 5        |
| 4.2. Desarrollo de la idea y correctitud. . . . .     | 5        |
| 4.3. Análisis de complejidad. . . . .                 | 5        |
| 4.4. Experimentación y gráficos. . . . .              | 5        |
| 4.4.1. Test 1 . . . . .                               | 5        |
| 4.4.2. Test 2 . . . . .                               | 5        |
| <b>5. Problema 3: Refinando petróleo</b>              | <b>6</b> |
| 5.1. Descripción del problema. . . . .                | 6        |
| 5.2. Desarrollo de la idea y correctitud. . . . .     | 6        |
| 5.3. Análisis de complejidad. . . . .                 | 6        |
| 5.4. Experimentación y gráficos. . . . .              | 6        |
| 5.4.1. Test 1 . . . . .                               | 6        |
| 5.4.2. Test 2 . . . . .                               | 6        |
| 5.4.3. Test 3 . . . . .                               | 6        |
| <b>6. Apéndice 1: acerca de los tests</b>             | <b>7</b> |
| <b>7. Apéndice 2: secciones relevantes del código</b> | <b>8</b> |
| 7.1. Código del Problema 1 . . . . .                  | 8        |
| 7.2. Código del Problema 2 . . . . .                  | 8        |
| 7.3. Código del Problema 3 . . . . .                  | 8        |

## **1. Objetivos generales**

## **2. Plataforma de pruebas**

Para toda la experimentación se utilizará un procesador Intel Atom, de 4 núcleos a 1.6 GHZ.  
El software utilizado será Ubuntu 14.04, y G++ 4.8.2.

### 3. Problema 1: Dakkar

#### 3.1. Descripción del problema.

Tenemos la intención de competir en una versión particular del Rally Dakkar, y para esto, decidimos usar nuestros conocimientos en computación a nuestro favor. Sabemos que en esta competencia, podremos usar una BMX, una motocross y un buggy arenero. Nuestro objetivo es finalizar la competencia en el menor tiempo posible para nuestras habilidades, y contamos con los siguientes datos:

- El circuito se divide en  $n$  etapas (numeradas del 1 al  $n$ ) y en cada etapa solo se puede usar un vehículo
- Para cada etapa, se sabe cuanto tardaría en recorrerla con cada vehículo
- Tanto la motocross como el buggy arenero tiene un número limitado de veces que se pueden usar, a diferencia de la BMX, y estos números son conocidos
- La complejidad del algoritmo pedido es de  $\mathcal{O}(n \cdot k_m \cdot k_b)$  donde  $n$  es la cantidad de etapas,  $k_m$  es la cantidad máxima de veces que puedo usar la moto, y  $k_b$  la cantidad máxima de veces que puedo usar la moto.

#### 3.2. Desarrollo de la idea.

#### 3.3. Demostración de correctitud.

#### 3.4. Análisis de complejidad.

#### 3.5. Experimentación y gráficos.

##### 3.5.1. Test 1

##### 3.5.2. Test 2

## 4. Problema 2: Zombieland II

### 4.1. Descripción del problema.

Luego del éxito rotundo del último ataque contra los zombies producido por nuestro anterior algoritmo, la humanidad ve un rayo de esperanza. En una de las últimas ciudades tomadas por la amenaza Z, se encuentra un científico que afirma haber encontrado la cura contra el zombieismo, rodeado por una cuadrilla de soldados del más alto nivel. Nuestro noble objetivo es entonces, proveer del camino más seguro (el que genere menos pérdidas humanas) al científico y sus soldados de manera tal que logren llegar desde donde están, hasta el bunker militar que se encuentra en esa ciudad. Para esto, se conocen los siguientes datos:

- La ciudad en cuestión tiene la forma clásica de grilla rectangular, compuesta por  $n$  calles paralelas en forma horizontal,  $m$  calles paralelas en forma vertical dando así una grilla de manzanas cuadradas. Los números  $n$  y  $m$  son conocidos.
- Se conoce la ubicación del científico y del bunker.
- Se conoce la cantidad de soldados que el científico tiene a su disposición.
- Si todos los soldados perecen, el científico no tiene posibilidad de sobrevivir por sí mismo (o sea, no puede llegar al bunker con 0 soldados).
- Se sabe la cantidad de zombies ubicados en cada calle.
- Si bien nuestros soldados tienen un alto nivel de combate, el enfrentamiento que tuvieron en el último ataque contra los zombies, acabaron con todas sus municiones por lo cual solo tienen sus cuchillos de combate. Esto incide entonces en que el grupo solo pasara por una calle sin sufrir pérdidas, si hay hasta un soldado por zombie, al menos, y en caso contrario, perderá la diferencia entre la cantidad de soldados, y la cantidad de zombies. En otras palabras, sea  $z$  la cantidad de zombies de una calle, y  $s$  la cantidad de soldados de que tiene la cuadrilla al momento de pasar por esa calle, si  $s \geq z$  entonces la cuadrilla pasa sin pérdidas. En caso contrario la cuadrilla pierde  $z - s$  soldados.
- Puede no existir un camino que asegure la supervivencia del científico.
- La complejidad del algoritmo pedido es de  $\mathcal{O}(s \cdot n \cdot m)$ .

### 4.2. Desarrollo de la idea y correctitud.

### 4.3. Análisis de complejidad.

### 4.4. Experimentación y gráficos.

#### 4.4.1. Test 1

#### 4.4.2. Test 2

## 5. Problema 3: Refinando petróleo

### 5.1. Descripción del problema.

Tenemos en cierta zona, pozos de petróleos que necesitan ser refinados. Para que un pozo pueda refinar su petróleo, necesita o bien una refinería ubicada junto a el, o bien debe conectarse mediante tuberías a otro pozo que o tenga una refinería, o este conectado mediante tuberías a algun pozo que puede refinar su petróleo. Nuestro objetivo es armar un plan de construcción que decida donde construir una refinería y donde construir un sistema de tuberías, de manera tal que todos los pozos puedan refinar su petróleo. Para esto, se tienen los siguientes datos:

- Se conoce la cantidad de pozos en cuestión.
- Se conoce el costo de construir una refinería (este costo es fijo, y no varía según el pozo).
- Dada la geografía del lugar, no todo par de pozos se puede comunicar por una tubería directamente, y por decisiones de administración, una tubería no puede bifurcarse a mitad de camino entre un pozo y otro. Igualmente, conocemos todos los pares de pozos que pueden conectarse con una tubería, y el costo de esta (a diferencia de las refinerías, las tuberías dependen del par de pozos que se quiere conectar).
- La complejidad del algoritmo debe ser estrictamente menor a  $\mathcal{O}(n^3)$ .

### 5.2. Desarrollo de la idea y correctitud.

### 5.3. Análisis de complejidad.

### 5.4. Experimentación y gráficos.

#### 5.4.1. Test 1

#### 5.4.2. Test 2

#### 5.4.3. Test 3

## **6. Apéndice 1: acerca de los tests**

## **7. Apéndice 2: secciones relevantes del código**

En esta sección, adjuntamos parte del código correspondiente a la resolución de cada problema que consideramos más **relevante**.

### **7.1. Código del Problema 1**

### **7.2. Código del Problema 2**

### **7.3. Código del Problema 3**