



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Algoritmos y Estructuras de Datos III
Primer Cuatrimestre de 2015

Integrante	LU	Correo electrónico
Aldasoro Agustina	86/13	agusaldasoro@gmail.com
Noriega Francisco	660/12	frannoriega.92@gmail.com
Zimenspitz Ezequiel	155/13	ezeqzim@gmail.com
Zuker Brian	441/13	brianzuker@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>

Poner resumen.

Índice

1. Dakkar	3
1.1. Descripción de la problemática	3
1.2. Resolución propuesta y justificación	4
1.3. Análisis de la complejidad	4
1.3.1. Complejidad Temporal	4
1.3.2. Complejidad Espacial	4
1.4. Código fuente	4
1.5. Experimentación	4
1.5.1. Contrastación Empírica de la complejidad	4
2. Zombieland II	5
2.1. Descripción de la problemática	5
2.2. Resolución propuesta y justificación	5
2.3. Análisis de la complejidad	5
2.4. Código fuente	5
2.5. Experimentación	5
2.5.1. Contrastación Empírica de la complejidad	5
3. Refinando petróleo	6
3.1. Descripción de la problemática	6
3.2. Resolución propuesta y justificación	6
3.3. Análisis de la complejidad	6
3.4. Código fuente	6
3.5. Experimentación	6
3.5.1. Contrastación Empírica de la complejidad	6

1. Dakar

1.1. Descripción de la problemática

La problemática trata de una travesía, la cual cuenta con n cantidad de etapas. Para cada una de las etapas, se puede elegir recorrerla en alguno de los tres vehículos disponibles: una BMX, una motocross o un buggy arenero. Cada uno de ellos permite concretar cada etapa en cantidades de tiempo diferentes. Además, la cantidad de veces que se pueden usar la motocross y el buggy arenero está acotada por k_m y k_b respectivamente.

Los *tiempos* que le llevan a los vehículos recorrer el trayecto varían por cada etapa y son datos conocidos pasados por parámetro.

Se pide recorrer la travesía, dentro de las restricciones, de modo que se utilice la menor cantidad de tiempo posible. Si existen dos (o más) maneras de atravesarla dentro del tiempo óptimo, se pide devolver sólo una.

Se exige resolver la problemática con una complejidad temporal de $O(n.k_m.k_b)$.

Dibujitos con ejemplos :)

1.2. Resolución propuesta y justificación

Para resolver esta problemática, optamos por implementar un algoritmo de *Programación Dinámica*.

Formulación Recursiva

$$func(n, k_m, k_b) = \begin{cases} 0 & \text{si } n = 0 \\ tiempoBici(n) + f(n-1, 0, 0) & \text{si } k_m = 0 \wedge k_b = 0 \\ \min \left(\begin{array}{l} tiempoBici(n) + func(n-1, 0, k_b), \\ tiempoBoogy(n) + func(n-1, 0, k_b-1) \end{array} \right) & \text{si } k_m = 0 \wedge k_b \neq 0 \\ \min \left(\begin{array}{l} tiempoBici(n) + func(n-1, k_m, 0), \\ tiempoMoto(n) + func(n-1, k_m-1, 0) \end{array} \right) & \text{si } k_m \neq 0 \wedge k_b = 0 \\ \min \left(\begin{array}{l} tiempoBici(n) + func(n-1, k_m, k_b), \\ tiempoMoto(n) + func(n-1, k_m-1, k_b), \\ tiempoBoogy(n) + func(n-1, k_m, k_b-1) \end{array} \right) & \text{sino} \end{cases}$$

1.3. Análisis de la complejidad

1.3.1. Complejidad Temporal

1.3.2. Complejidad Espacial

Si bien, ya no piden ningun requisito, pongamos cuanta memoria usa :)

1.4. Código fuente

1.5. Experimentación

1.5.1. Constrastación Empírica de la complejidad

2. Zombieland II

2.1. Descripción de la problemática

2.2. Resolución propuesta y justificación

2.3. Análisis de la complejidad

2.4. Código fuente

2.5. Experimentación

2.5.1. Contrastación Empírica de la complejidad

3. Refinando petróleo

3.1. Descripción de la problemática

3.2. Resolución propuesta y justificación

3.3. Análisis de la complejidad

3.4. Código fuente

3.5. Experimentación

3.5.1. Contrastación Empírica de la complejidad