

# Problemas de Diseño y Análisis de Algoritmos

## Problema 2: Electricidad

**Equipo: 3**

**Miembros:**

- Laura Brito Guerrero
- Sheyla Cruz Castro
- Rodrigo García Gómez

### **Enunciado del Problema:**

Fito tiene un trabajo interesante y muy bien remunerado. Cada mañana él debe apagar las lámparas de las calles de su barrio, que casualmente están todas en un mismo lado de una vía recta. Fito, que gusta de divertirse en exceso, está de fiesta todas las noches hasta las 05 : 00 AM y es exactamente a esa hora que arranca su faena de apagado de luces. Cada lámpara tiene un consumo definido y como Fito es muy responsable con el ahorro de la electricidad, su meta es apagar las lámparas en el orden en que se minimice el total de electricidad gastada. Fito se mueve a una velocidad de  $1\text{ m/s}$  y se sabe que apagar una lámpara no toma tiempo extra por lo cual Fito puede hacerlo cuando pasa por su lado. Conociendo la posición inicial de Fito (que siempre es al lado de una de las lámparas), el consumo de cada una de estas y su respectiva ubicación, desarrolle un algoritmo que ayude a Fito a concretar su objetivo.

### **Abstract**

Para este problema se logró obtener un algoritmo *fuerza bruta* el cual es exponencial ya que revisa todos los subconjuntos posibles de caminos. Se siguieron estrategias *greedy*s en tiempo polinomial pero eran incorrectas, por esta razón se cree que el problema es NP-completo e igualmente no llegamos a ninguna demostración, tratamos de reducirlo en los problemas del *Viajante*, el 3-SAT y el *Camino de Hamilton* sin obtener resultados satisfactorios.

### **Idea Fuerza Bruta:**

**(Brute1)**

El algoritmo recibe como entrada: Un arreglo de posiciones de tamaño  $n$  (siendo  $n$  la cantidad de farolas a apagar) que tendrá en la posición  $i$ , un entero que representa la

posición de la farola  $i$ . Se asume que el arreglo está ordenado de forma creciente. Un arreglo de costos de tamaño  $n$ , que tendrá en la posición  $i$ , el costo de electricidad por segundo de la farola en la posición  $i$  del arreglo de posiciones. Un entero que representa la posición inicial de Fito.

Primeramente se abordó el problema con un enfoque de *Fuerza Bruta*, intentando generar todos los caminos que Fito es capaz de hacer, para luego compararlos y quedarse con el de menor consumo de electricidad. La idea es, al estar Fito posicionado en  $k$ , comparar recursivamente la mejor solución al avanzar hacia la derecha con la mejor al avanzar hacia la izquierda en busca de la farola encendida más cercana. En caso de estar apagadas todas las farolas en una dirección, se avanzará hacia la otra. Al estar apagadas las farolas en ambas direcciones, se entiende que se alcanzó un camino en el que todas las farolas se apagaron, y se retornará cero en esa iteración. Para conocer la condición de las farolas en cada momento, se lleva un arreglo booleano de tamaño  $n$  llamado *match*. Al entrar en cada llamado al método recursivo, se marcará como apagada la posición actual de Fito, y al retornar, se volverá a encender. Luego con dos ciclos, uno de  $k$  hasta  $n$ , y otro de  $k$  hasta 0, se buscarán dichas farolas encendidas más próximas (es evidente que entre ambos ciclos se hace como máximo  $n$  iteraciones). Además, se llevará una variable con el costo actual por segundo de las farolas encendidas, el cual disminuirá al inicio de cada llamado recursivo en una cantidad igual al costo de la farola de la posición actual de Fito (esto representa la acción de apagar dicha farola). Luego, el costo de la opción óptima hacia la izquierda y el de la opción óptima hacia la derecha, será igual a la diferencia modular de la posición actual con la posición izquierda o derecha multiplicado por el costo actual por segundo de las farolas, y sumado con el llamado recursivo al método, pasando  $k$  como la posición izquierda o derecha respectivamente. El método retornará el menor costo entre las opciones antes mencionadas.

Al estar Fito en una posición, tendrá solo dos opciones de movimiento, ya que las farolas están posicionadas linealmente. Como el algoritmo analiza, para cada posición, todas las opciones posibles, es evidente que todas las opciones de movimiento de Fito son analizadas, y por tanto, siempre se encontrará la óptima. El algoritmo crece exponencialmente, puesto que se analizarán siempre todas las  $k$  permutaciones coherentes de farolas para toda  $k$  entre 1 y  $n - 1$  (es solo hasta  $n - 1$  puesto que la farola inicial siempre ocupará el primer lugar de las permutaciones de caminos). Sin embargo, el costo del algoritmo es considerablemente menor a lo antes mencionado, ya que no se analizan las numerosas permutaciones incoherentes en las que Fito pasa sobre las farolas sin apagarlas. Estas son las que, para algún  $i, j, i < j < k$ , la farola de la posición  $i$  esté antes que la de la posición

$j$  o para  $k < i < j$ , la farola de la posición  $j$  esté antes que la de la posición  $i$ . Al analizar esto, el equipo se percató de una posible mejora.

### **Idea 2:**

Al encontrarse en una farola, cuyas farolas izquierdas o derechas estén todas apagadas, es posible avanzar directamente hacia la última farola en la dirección contraria, apagando todas las farolas intermedias, sin necesidad de analizar nunca el lado oscuro. Esto es posible hacerlo con solo un ciclo lineal. Por desgracia esto no resultó ser más que un algoritmo aún más lento que el anterior. Sin embargo, la idea se mantuvo en mente para el futuro.

### **Idea 3:**

De esa forma surgió la idea de implementar un algoritmo dinámico, la idea se basa en el hecho de que empezando por las esquinas, solo hay un camino posible, y por tanto este será el óptimo. Así se calcula, empezando por la esquina más cercana a la posición inicial, lo más óptimo comenzando por cada farola hasta llegar a la posición inicial de Fito, utilizando lo más óptimo de la farola anterior ya calculada. El problema de este método está en que la solución óptima de cada farola anterior, varía mucho al estar la nueva farola a calcular, apagada desde el inicio. Por tanto, la idea fue descartada.

### **Idea 4:**

Ante la ausencia de mejores ideas, se desarrolló una *fuerza bruta mejorada*. Al completar el primer camino entero con el método recursivo, su costo puede ser almacenado en una variable global. De esta manera antes de continuar con la implementación de cada método recursivo, es posible comprobar si la suma actual es ya mayor que la solución almacenada en dicha variable. En caso negativo, es evidente que no es necesario seguir con ese llamado y se puede retornar instantáneamente. En caso de finalizar otro camino entero, se compara y actualiza la variable en caso de ser necesario. Nótese que el algoritmo sigue teniendo complejidad  $2^n$ , sin embargo, potencialmente se reducirá mucho el tiempo de ejecución, pues imaginando las opciones de camino como un árbol binario, es posible *cortar* ramas de grandes tamaños, las cuales se dejarían de revisar.

### **Idea 5:**

(greedy)

A continuación se utilizó un enfoque *greedy* para intentar resolver el problema. Estando Fito posicionado en una farola, querrá apagar prioritariamente, las farolas de mayor consumo y las farolas más cercanas a él para ahorrar el consumo de estas. Por tanto, se hace una razón de costo de farola  $i$  dividida entre la distancia entre las farolas  $i$  y  $k$ . Sin embargo al compararlo con los resultados del algoritmo de *fuerza bruta* fue evidente que el algoritmo era incorrecto para ciertos casos. Para lograr una comprobación bastante acertada, se realizó un generador aleatorio de casos de prueba que genera una gran cantidad de forma automática.

### **Idea 6:**

Siguiendo una estrategia *greedy*, debemos seleccionar cual es la mejor opción en el paso  $i$ -ésimo. Se comienza en una posición  $k$  y se lleva constancia de las lámparas que se encuentran encendidas a la izquierda y las encendidas a la derecha.

En cada paso decidimos si ir a la izquierda o hacia la derecha, se lleva una suma de los costos de las lámparas que quedan encendidas, y se halla el gasto de moverse a la lámpara encendida a la izquierda y se comparará con el gasto total de moverse a la lámpara encendida a la derecha, se decide moverse hacia la lámpara que maximice los gastos en ese paso.

Esta estrategia *greedy* es incorrecta, ofrece resultados correctos en muchos escenarios posibles. El fallo está en que no siempre es mejor elegir la lámpara que maximice los gastos en el paso actual aunque parece correcta ya que trata de que la parte donde más consumo puede ser obtenido sea apagada lo antes posible, esto no siempre minimiza los gastos porque puede ocurrir que luego de un paso para apagar la lámpara que más consumo permite en los pasos posteriores aumente su distancia de otra que pueda aumentar considerablemente ese gasto.