



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO

Ejercicio 08: Diseño de soluciones con  
Programación Voraz

Unidad de aprendizaje: Análisis de Algoritmos

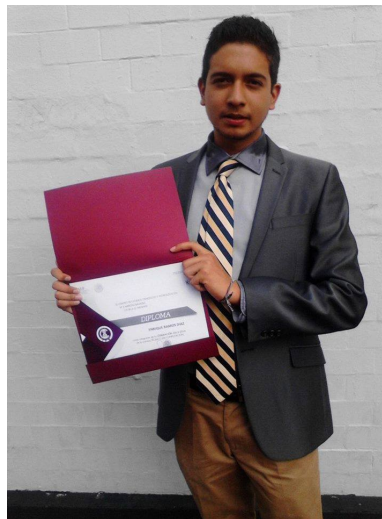
Grupo: 3CM3

*Alumno:*

Ramos Diaz Enrique

*Profesor(a):*

Franco Martínez Edgardo Adrián



19 de noviembre 2018

# Índice

<b>1</b>	<b>Andar Chido Reloaded</b>	<b>1</b>
1.1	Problema	1
1.2	Análisis y solución	1
1.3	Código resultante	2
1.4	Validación del juez online	4

## 1. Andar Chido Reloaded

### 1.1. Problema

El Dr. Ethan Freud, famoso por su "Estudio de los Azucarados" y el "Análisis del Dinero Infinito", se dio cuenta de la tremenda oportunidad de negocios que representa regresar a sus casas a la gente que terminó "un poco" mal gracias a "un poco" de exceso de bebidas "espirituales" altas en "azúcar".

Ahora el Dr. Ethan Freud tiene  $N$  clientes en su oficina que le pagarán por llevarlos a sus casas. Para cada cliente  $i$ , le toma  $T_i$  minutos llegar desde su oficina hasta la casa del cliente. Sin embargo, sólo puede llevar a un cliente a la vez, por lo que debe dejar a los demás esperando. El problema es que como están un poco "altos en azúcar", cada cliente  $j$  que se quedó destruye  $D_j$  de sus preciados libros cada minuto. Entonces, mientras le toma  $2T_i$  minutos dar la vuelta en llevar al cliente  $i$ , los clientes destruyeron entre todos  $2T_i S$  libros, donde  $S$  es la suma de las  $D_j$  de los clientes esperando.

Como el Dr. Ethan Freud no quiere que rompan muchos de sus libros te ha pedido que le ayudes a saber la menor cantidad de libros destruidos posibles dada la información de los clientes de esta noche.

#### Entrada

En la primera línea, un entero:  $N$ .

En las siguientes  $N$  líneas, dos enteros:  $T_i$  y  $D_i$  del  $i$ -ésimo cliente.

#### Salida

La mínima cantidad de libros destruidos después de llevar a todos los clientes a sus casas.

### 1.2. Análisis y solución

La solución por fuerza bruta de este problema es revisar todas las posibles combinaciones de formas en las que se podrían llevar a los clientes a sus casas, calcular el total de libros destruidos y hallar el menor de ellos, pero la complejidad se dispararía hasta un orden de  $O(c^n)$ .

Primero hay que calcular el tiempo de ida y vuelta  $2T_i$  por cada uno de los clientes, que sería el doble del tiempo que se ingresa, y además calcular todos los posibles  $S$  libros que serán destruidos, sumando los libros destruidos por cada cliente.

Ahora bien, lo que realmente importa aquí es el orden en que el doctor llevará los clientes a sus casas, por lo que necesitamos un algoritmo de ordenación. Utilizamos el algoritmo de ordenamiento Shell para esto. Posteriormente debemos definir la condición que determinará si hay que intercambiar

o no a un cliente de lugar en la "fila de espera", de modo que los primeros clientes en irse sean los que destruyen más libros y los que esperen, menos.

Pueden existir dos casos:

1. Si primero atendemos al cliente  $i$ , el número total de libros  $X$  que destruye el cliente  $i - k$  esperando será el número de libros que destruye por minuto  $D_{i-k}$  por el tiempo de trayecto total  $2T_i$  que tarda en irse el cliente  $i$ . Es decir:

$$X = \text{clientes}[i - k].d * \text{clientes}[i].t$$

2. Si primero atendemos al cliente  $i - k$ , el número total de libros  $Y$  que destruye el cliente  $i$  esperando será el número de libros que destruye por minuto  $D_i$  por el tiempo de trayecto total  $2T_{i-k}$  que tarda en irse el cliente  $i - k$ . Es decir:

$$Y = \text{clientes}[i].d * \text{clientes}[i - k].t$$

Si el cliente  $i - k$  destruye menos libros que el cliente  $i$  en su espera (caso 2), los intercambiamos. Debemos atender primero a quienes destruyan más libros. Es decir:  $X > Y$ , intercambiamos.

Una vez obtenido el orden, de forma que minimizamos el total de libros destruidos, vamos "atendiendo" a los clientes, acumulando los libros destruidos de los clientes que esperan mientras se atiende al actual.

Este valor será el mínimo de todas las combinaciones.

### 1.3. Código resultante

La complejidad de este algoritmo es la misma que la del método de ordenamiento Shell,  $O(N \log_2 N)$ , pues lo demás son recorridos lineales sobre el arreglo de  $N$  clientes.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  //Estructura del cliente
5  typedef struct{
6      long long int t; //Tiempo de trayecto
7      long long int d; //Libros que destruye
8  } CLIENTE;
9
10 void MinLibros(CLIENTE c[], long long int n){
11     //Debemos establecer el orden en que se regresaran a los clientes
12     int k = trunc(n/2), b = 0;
13     double tempt = 0, tempd = 0;
14     //Utilizamos el algoritmo shell de ordenamiento
15     while(k >= 1) {

```

```

16     b = 1;
17     while(b != 0){
18         b = 0;
19         for(int i = k; i <= (n-1); i++){
20             /*Vamos revisando si un cliente destruye más libros que otro,
21             segun el tiempo que tarde en llevar al otro. De mayor a menor
22             números de libros destruidos
23             c[i] Cliente i: Destruye X libros en el tiempo del cliente i-k
24             c[i-k] Cliente i-k: Destruye Y libros en el tiempo del cliente i*/
25             if(c[i].d * c[i-k].t > c[i-k].d * c[i].t){
26                 /*Si el cliente i destruye más libros que el cliente i-k,
27                 los intercambiamos*/
28                 tempt = c[i].t;
29                 tempd = c[i].d;
30                 c[i].t = c[i-k].t;
31                 c[i].d = c[i-k].d;
32                 c[i-k].t = tempt;
33                 c[i-k].d = tempd;
34                 b = b + 1;
35             }
36         }
37     }
38     k = trunc(k/2);
39 }
40 }
41 //N = numero de clientes en la oficina
42 //Ti = minutos en llegar de la oficina a la casa del cliente
43 //Dj = libros destruidos por minuto por cada cliente
44 //2Ti = minutos de ida y vuelta por cliente
45 //S = suma de las Dj de los N cliente esperando
46 int main(){
47     long long int N = 0, S = 0, libmin = 0;;
48     scanf("%lli", &N);
49     CLIENTE clientes[N];
50     for(int i = 0; i < N; i++){
51         scanf("%lld %lld", &clientes[i].t, &clientes[i].d);
52         //Minutos totales de ida y regreso por cliente
53         clientes[i].t = clientes[i].t * 2;
54         //Suma de las Dj de los clientes esperando
55         S = S + clientes[i].d;
56     }
57     MinLibros(clientes, N);
58     /*Ya ordenados los clientes de forma que se minimizan los libros
59     destruidos, atendemos a los clientes*/
60     for (int i = 0; i < N; i++){
61         /*Le restamos al total los libros que destruiría el cliente

```

```

62 que se atiende*/
63     S = S - clientes[i].d;
64 /*Vamos acumulando cuantos libros destruyen los otros clientes
65 mientras se atiende al cliente actual*/
66     libmin = libmin + S * clientes[i].t;
67 }
68 printf("\n%lld", libmin);
69 return 0;
70 }

```

## 1.4. Validación del juez online

https://omegaup.com/arena/problem/Andar-Chido-Reloaded#problems

[Arena](#)
[Concursos](#)
[Problemas](#)
[Rank](#)
[Escuelas](#)
[Blog](#)
[Preguntas](#)

Entrada	Salida	Descripción
<pre> 6 3 1 2 5 2 3 3 2 4 1 1 6 </pre>	86	<p>El Dr. Ethan Freud regresa a sus clientes en el siguiente orden: 6, 2, 3, 4, 1, 5. Mientras lleva al cliente 6 a su casa, el resto destruye 24 libros. Mientras lleva al cliente 2, pierde otros 28 libros. Para los clientes 3, 4, y 1, el resto destruye 16, 12, y 6 libros, respectivamente. Mientras lleva al cliente 5 ya no hay clientes que destruyan los libros en la oficina. El total de libros destruidos es 86.</p>

**Límites**

- $1 \leq T_i \leq 2 \cdot 10^6$
- $1 \leq D_i \leq 100$

**Subtareas**

- Para 14% de los puntos,  $2 \leq N \leq 10$
- Para el siguiente 19% de los puntos,  $2 \leq N \leq 21$
- Para el siguiente 28% de los puntos,  $2 \leq N \leq 1000$
- Para el último 39% de los puntos,  $2 \leq N \leq 10^5$

Fuente: Freddy  
 Problema subido por: [Freddy Román Cepeda](#)  
[Reportar contenido inapropiado en este problema.](#)

**Envíos**

Enviado	GUID	Estatus	Porcentaje	Lenguaje	Memoria	Tiempo	Detalles
2018-11-18 13:49:03	<a href="#">03985d35</a>	Respuesta correcta	100.00%	c	3.25 MB	0.33 s	
2018-11-18 13:45:10	<a href="#">f94f21b9</a>	Respuesta correcta	100.00%	cpp11	3.23 MB	0.31 s	