



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2133 - ESTRUCTURAS DE DATOS Y ALGORITMOS

Informe Tarea 0

20 de abril de 2020

1º semestre 2020 - Profesores Vicente Dominguez - Yadran Eterovic

Pablo Garnham Guzmán - 15638189

Estructuras

Primero, quiero explicar como estructuré mi tarea:

Creé una lista ligada, en la que cada nodo era un arreglo de enteros de largo 2000. Por lo que para acceder al elemento de índice i , sólo debía dividir por 2000 y usar la función techo. De esta manera, quería evitar tener que recorrer cada vez la lista completa para acceder a un elemento, y en vez de eso, ir saltando de a 2000 elementos.

Luego, decidí que cada puerta iba a tener 3 de estas listas híbridas (pues es una mezcla entre arreglos y listas ligadas). Una lista para niños, otra para adultos, y otra para robots. De esta manera, era más fácil calcular que puerta era más conveniente cuando llegaba un pasajero. Por otro lado, guardaba el largo como parte del struct, de manera que no tenía que calcularlo cada vez, sino ir sumando 1 cada vez que se agregaba a alguien. Por otro lado, decidí que al hacer un abordaje, no iba a hacer ningún cambio a la lista, sino que iba a ir desplazando virtualmente la lista, de manera que sólo debía calcular cuantos niños, adultos y robots se iban en cada caso, y acumular este valor en el struct. De esta manera, al acceder a las listas, debía sumar este valor guardado anteriormente al índice que se buscaba, ignorando la parte inicial de la fila, que virtualmente ya se había evacuado.

Por otro lado, en el caso de los láser, lo que hacía, para evitar errores, esta vez decidí crear una nueva lista para el tipo de pasajero (niño, adulto o robot) al que le cayera el láser.

El cierre de una puerta, simplemente recorría las tres listas correspondientes a esa puerta, y aplicaba la función de ingreso, que no tiene mayor complejidad.

Finalmente la clausura, la más compleja de mis operaciones, consistía en calcular cual era la puerta con la mayor cantidad de pasajeros dentro de ese terminal, y luego crear un nuevo terminal auxiliar con esa cantidad de puertas. Luego iba en orden. En la primera puerta estaban los primeros de cada puerta original, en la segunda estaban los segundos, y así hasta

acabar. Luego creaba un último terminal auxiliar, esta vez con solo una puerta. Luego unía todas las filas de niños en una sola, siguiendo el orden de índices original, y hacía lo mismo con adultos y robots. Finalmente, hacía el ingreso en orden de cada uno de los integrantes de esta única puerta auxiliar.

Estas explicaciones de los algoritmos que usé pueden verse reflejados claramente en los cálculos que posteriormente se anuncian, y cómo los más simples son el abordaje y el láser, y el más complejo con mucho es la clausura de un terminal, en el que se privilegió la correctitud antes que la eficiencia. Todo esto considerando siempre el peor caso.

A continuación se hace el cálculo para cada uno de estos procedimientos:

INGRESO

En palabras simples, lo que hace esta función es

Para realizar el ingreso de una persona, realicé la función `choice_ingreso()` que ejecutaba aproximadamente 7 instrucciones con $O(1)$ y luego un ciclo tipo **for** con 5 instrucciones en su interior. Este ciclo dependía de la cantidad de puertas del terminal en cuestión.

Luego, se ejecutaba la función para agregar un elemento a la lista correspondiente. Esta, en el peor caso, ejecutaba a su vez la función para extender la lista (lo que ocurría con cierta periodicidad). Esta última función tenía 9 instrucciones en su interior, sin mayor complejidad. Luego habían otras 5 instrucciones simples para terminar de agregar el elemento a la lista.

Finalmente, la complejidad sería:

Si p es el número de puertas:

$$5p + 7 + 9 + 5$$

$$5p + 21 \in O(p)$$

Si consideramos que esta función se repetía para cada pasajero, y consideramos los pasajeros como n :

$$O(\text{INGRESO}) \in O(np)$$

ABORDAJE

La función de abordaje es efectuada por la función `escape`. Este incluía 13 instrucciones simples, y un ciclo **for** que realizaba 8 ciclos.

Dentro de este, se hacía una llamada a la función `gate_access()` que tenía 6 instrucciones simples, y una llamada a la función `list_acces()`. Esta última tenía un ciclo **for**, que dependía de la cantidad de pasajeros que tuviera aquella lista, específicamente, la cantidad de pasajeros dividida en 2000 (si, dos mil).

Por lo tanto, resulta:

$$\begin{aligned} &13 + 8(6 + 4 + \lceil n/2000 \rceil) \\ &= 13 + 48 + 32 + 8\lceil n/2000 \rceil \\ &O(\text{ABORDAJE}) \in O(n) \end{aligned}$$

CIERRE

En este caso, uso la función `cierre()`, la cual tiene 3 instrucciones simples, y un ciclo **for**, el que consiste en una llamada a la función `gate_acces()` y otra llamada a la función `choice_ingreso()`. La primera ya fue explicada en el procedimiento de ABORDAJE, y la segunda corresponde al procedimiento completo de INGRESO. Por lo tanto, la complejidad consiste en:

$$3 + n(6 + 4 + \lceil n/2000 \rceil + np)$$

$$3 + 6n + 4n + n^2/2000 + n^2p$$

$$O(\text{CIERRE}) \in O(n^2p)$$

CLAUSURA

Esta sección parte con un ciclo que recorre todas las puertas, para calcular la puerta con más gente. Llamemos p a la cantidad de puertas, y n a la cantidad de personas máxima que hay en una de estas puertas.

Luego, creamos un terminal auxiliar, para lo que se hace un **for** que efectúa n ciclos.

Luego, vamos llenando este terminal, por lo que hacemos un doble **for**:

El primero recorre n veces, el interior realiza np ciclos. En este último ciclo, se ejecutan las funciones `gate_access()` y `list_add()`, analizadas anteriormente.

Luego, al salir de este doble ciclo, se realiza otro doble **for**. La primera sección realiza n ciclos, y la interior nuevamente realiza np ciclos. En el interior se ejecutan las instrucciones `list_access()` y `list_add()`, también mencionadas y analizadas anteriormente.

Finalmente, se realiza otro ciclo **for**, esta vez realiza np ciclos, y sus instrucciones interiores son `list_access()` y `list_add()`.

Por lo tanto, finalmente la complejidad resulta:

$$p + n + np(\lceil n/2000 \rceil + O(1)) + n^2p(\lceil n/2000 \rceil + O(1)) + np(\lceil n/2000 \rceil + np)$$

Desarrollando y simplificando, tenemos que:

$$p + n + 2n^2p/2000 + np + n^3p/2000 + n^2p + n^2p^2$$

Luego, considerando sólo los términos de mayor grado, obtenemos que:

$$= O(\text{CLAUSURA}) \in O(n^3p + n^2p^2)$$

LÁSER

Esta maniobra la llevaba a cabo la función `laser_killer()`, la que realizaba varias instrucciones $O(1)$, y luego llamaba a la función `laser_shift()`, la que creaba una nueva lista en $O(1)$ y luego recorría la lista antigua agregando cada elemento. Por lo tanto:

$$O(\text{LÁSER}) \in O(n)$$

Siendo n la cantidad de pasajeros que tiene la puerta a la que llega el láser.