Algoritmos y estructuras de datos III

Trabajo Práctico Nº1

Carla Livorno 424/08 carlalivorno@hotmail.com
Completen!!!

Abril 2010

Índice

Introducción

Este trabajo tiene como objetivo la aplicación de diferentes técnicas algorítmicas para la resolución de tres problemas particulares, el cálculo de complejidad teórica en el peor caso de cada algoritmo implementado, y la posterior verificación empírica.

El lenguaje utilizado para implementar los algoritmos de todos los problemas fue C/C++

1. Problema 1

Enunciado del problema...

1.1. Explicación

Poner algo aca...

1.1.1. Análisis de complejidad

Y aca..

1.2. Detalles de la implementación

1.3. Pruebas y Resultados

2. Problema 2

2.1. Explicación

2.1.1. Análisis de complejidad

2.2. Detalles de la implementación

2.3. Pruebas y Resultados

3. Ejercicio 3

3.1. Explicación

Para cada instancia tenemos una lista que contiene para cada programador su horario de ingreso a la empresa y otra con su horario de egreso. Además, tenemos guardado en cada momento la cantidad máxima de programadores en simultáneo.

Dado que ambas listas se encuentran ordenadas, nuestro algoritmo las recorre decidiendo a cada momento si se produce un ingreso o un egreso, es decir, si el horario que sigue en la lista de ingresos es anterior a la de egresos implica que hubo un ingreso, en caso contrario un egreso.

Cuando una persona ingresa a la empresa se incrementa el contador de la cantidad de programadores en simultáneo en el horario actual. Así, cuando se produce un egreso se compara si la cantidad de programadores dentro de la empresa previo a dicho egreso es mayor a la máxima cantidad de programadores en simultáneo hasta el momento, de ser así, actualizamos el máximo. Luego se descuenta el recientemente egresado del contador çantidad de programadores en simultáneo".

Este procedimiento se repite hasta haber visto todos los ingresos, lo que nos garantiza tener el máximo correspondiente, ya que a partir de ese momento sólo se producirían egresos.

3.1.1. Análisis de complejidad

Elegimos el modelo uniforme para analizar la complejidad de este algoritmo porque el tamaño de los elementos es acotado y por lo tanto todas las operaciones elementales son de tiempo constante.

Sea n la cantidad de programadores.

```
programadores_en_simultaneo(ingresos, egresos)
    max, tmp, j, k ← 0
    while (j < n)                                          O(n)
        if (ingresos[j] ≤ egresos[k])
            tmp ← tmp + 1
            j ← j + 1
        else
            if (tmp > max)
                max ← tmp
            tmp ← tmp − 1
            k ← k + 1
    if (tmp > max)
        max ← tmp
    return max
```

Cada programador tiene un ingreso y un egreso, por lo tanto la lista de ingresos y la lista de egresos tienen longitud n. El algoritmo recorre completamente la lista de ingresos. El peor caso es cuando el último ingreso y el último egreso corresponden al mismo programador, ya que para registrar éste último ingreso, la lista de egresos tuvo que ser recorrida hasta encontrarse con el último. Por este motivo, podemos inferir que a lo sumo se realizan $2n - 1$ iteraciones. En cada una de estas iteraciones tenemos un costo constante de operaciones, que no modifican la complejidad en el análisis asintótico. La complejidad algorítmica entonces, es $\mathrm{O}(n)$.

3.2. Detalles de la implementación

3.3. Pruebas y Resultados

4. Compilación y ejecución de los programas

Para compilar los programas se puede usar el comando `make` (Requiere el compilador `g++`). Se pueden correr los programas de cada ejercicio ejecutando `./bn_mod_n`, `./ronda_de_amigas` y `./programadores` respectivamente.

Los programas leen la entrada de stdin y escriben la respuesta en stdout. Para leer la entrada de un archivo `Tp1EjX.in` y escribir la respuesta en un archivo `Tp1EjX.out` ses puede usar:

```
./(ejecutable) <Tp1EjX.in >Tp1EjX.out
```

Para medir los tiempos de ejecución: `./(ejecutable) time`

Para contar la cantidad de operaciones: `./(ejecutable) count`. Devuelve la cantidad de operaciones de cada instancia.

5. Conclusiones