

Trabajo práctico 1

Primera entrega: 13 de abril de 2012, hasta las 18:00 horas.

Segunda entrega: 27 de abril de 2012, hasta las 18:00 horas.

Nota: El trabajo práctico se aprobará con la nota de 5 (*cinco*). Cada problema podrá ser recuperado individualmente.

En caso de recuperar o entregar en segunda fecha la nota final será dada por el 20 % del puntaje otorgado a los ejercicios en la primera fecha más el 80 % del puntaje otorgado a esos ejercicios en la segunda entrega.

Este trabajo práctico consta de 3 problemas. Para cada uno se pide encontrar una solución algorítmica al problema propuesto y luego desarrollar los siguientes puntos:

1. Explicar las ideas de forma clara, sencilla, estructurada y concisa. Para esto se puede utilizar pseudocódigo o lenguaje coloquial, o combinar ambas herramientas. Es importante que lo expuesto en este punto sea suficiente para el desarrollo de los puntos subsiguientes, pero no excesivo (no es un código).
2. Justificar por qué el procedimiento explicado en el primer punto resuelve efectivamente el problema.
3. Encontrar una cota de complejidad temporal del algoritmo propuesto. Los parámetros en función de los cuales dar la cota están detallados en el problema. Utilizar el modelo uniforme salvo que se explicita lo contrario.
4. Justificar por qué el algoritmo presentado en el primer punto cumple la cota de complejidad dada en el punto anterior.
5. Dar un código fuente claro que implemente la solución propuesta. El mismo no sólo debe ser correcto sino que además debe estar bien programado. Si bien no se pide que siga ninguna convención de codificación específica, mínimamente el mismo debe tener nombres de variables apropiados y un estilo de indentación coherente.
6. Dar un conjunto exhaustivo de casos de test que permitan corroborar en la práctica la correctitud del problema. Exhaustivo quiere decir que se debe explicar cómo los tests cubren los casos posibles del problema, **no** quiere decir que deben ser necesariamente muchísimos casos ni tampoco muy grandes, salvo que el tamaño sea una necesidad para determinar la correctitud.
7. Dar un conjunto de casos de test que permitan observar la performance en términos de tiempo del problema. Para esto se deben desarrollar tanto tests patológicos de peor caso como tests generados sin una intencionalidad.
8. Presentar en forma de gráfico la comparación entre tiempo estimado de corrida según la complejidad temporal calculada, tiempo medido de corrida para los tests patológicos de peor caso, y tiempo medido de corrida para los tests sin intencionalidad.

Respecto de las implementaciones, es aceptable cualquier lenguaje que permita el cálculo de complejidades según la forma vista en la materia. Además, debe correr correctamente en las máquinas de los laboratorios del Departamento de Computación. Java es el lenguaje sugerido por la Cátedra, en la versión instalada en dichos laboratorios. Se recomienda fuertemente consultar si se de duda si un lenguaje es aceptable o no.

La entrada y salida debe hacerse por medio de archivos. No se considerará correcta una implementación que no pase los tests que se mencionaron en los puntos anteriores.

Será valorada la presentación de un análisis de casos críticos para los problemas presentados, podas posibles de ser pertinentes y caminos alternativos que llevaron hasta la solución presentada.

Problema 1

Dado un vector v de enteros, se define meseta como un segmento de posiciones consecutivas con el mismo valor. Sean dos vectores X e Y no vacíos, cada uno ordenado en forma no decreciente. Hallar la longitud de la meseta más extensa que surja de combinar ambos vectores en uno resultante, también ordenado. Resolver el problema en $O(|X| + |Y|)$

Entrada Tp1Ej1.in

El archivo contiene varias instancias, cada una consta de dos líneas, una para cada vector con los enteros separados por un espacio en blanco.

Salida Tp1Ej1.out

Para cada instancia de entrada, se debe indicar una línea con la longitud de la meseta encontrada.

Problema 2

Sea una matriz de enteros de $n \times m$, no vacía, definimos meseta como el conjunto de celdas adyacentes sobre sus lados que comparten un mismo valor. Hallar la cantidad de elementos de la meseta más extensa. Resolver el problema en $O((n * m)^2)$

Entrada Tp1Ej2.in

El archivo contiene varias instancias, y cada una consta de dos líneas. La primera contiene dos enteros separados por un blanco para n y m respectivamente. La segunda contiene todos los elementos de la matriz, separados entre sí por un blanco, siguiendo el orden natural de lectura.

Salida Tp1Ej2.out

Para cada instancia de entrada, se debe indicar una línea con la cantidad de celdas que ocupa la meseta de mayor extensión.

Problema 3

Se dispone de un tablero de $n \times m$ en el que se colocan t fichas, en t casillas distintas. El objetivo del juego es ir eliminando las fichas mediante saltos hasta dejar sólo una.

Una ficha se saca del tablero cuando otra adyacente salta por sobre ella en sentido vertical u horizontal quedando en el casillero inmediato siguiente, que debía estar desocupado antes del salto.

Para cada configuración de fichas ofrecidas indicar, de ser posible, los pasos a seguir para eliminar todas las fichas menos una.

Entrada Tp1Ej3.in

El archivo contiene varias instancias, y cada una consta de dos líneas. La primera contiene n y m separados por un blanco. La segunda contiene una sucesión de t pares indicando las coordenadas x e y de las respectivas t fichas, usando blancos como separadores entre x e y y ';' como separador de las coordenadas.

La secuencia será de la forma: $\langle x \rangle \langle y \rangle [; \langle x \rangle \langle y \rangle]$ y donde: $\langle x \rangle$ e $\langle y \rangle$ son enteros

Salida Tp1Ej3.out

Para cada instancia de entrada, se debe indicar una línea con la serie de pasos necesarios para resolver el juego, o una línea en blanco si no es posible.

La secuencia de pasos será de la forma:

$\langle \text{coordenada de origen} \rangle; \langle \text{coordenada de destino} \rangle [; \langle \text{coordenada de origen} \rangle; \langle \text{coordenada de destino} \rangle]$

Donde: coordenada de origen: $\langle x \rangle \langle y \rangle$