

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
Primer Examen
(Segundo Semestre de 2013)

Horario 0581: prof. Andrés Melgar
 Horario 0582: prof. Fernando Alva

Duración: 3 horas

Nota:

- No se permite el uso de material de consulta.
- No se ofrecerá asesoría en la parte teórica.
- **La presentación, la ortografía y la gramática influirán en la calificación.**

Puntaje total: 20 puntos

Cuestionario:

PARTE TEÓRICA

Responda las siguientes preguntas según los conceptos vistos en clase:

Pregunta 1 (1 punto) ¿Cuál es la diferencia entre el algoritmo de inserción lineal y el de inserción binaria? **Justifique adecuadamente su respuesta.**

Pregunta 2 (2 puntos) El siguiente algoritmo ordena el arreglo a que posee n elementos. Se le pide que describa cuál es la estrategia de ordenación usada en el algoritmo. **No se pide el pseudocódigo del algoritmo.**

```

intervalo ←  $n \div 2$ 
while intervalo > 0 do
  for  $i = \text{intervalo} + 1$  to  $n$  do
     $j \leftarrow i - \text{intervalo}$ 
    while  $j > 0$  do
       $k \leftarrow j + \text{intervalo}$ 
      if  $a[j] \leq a[k]$  then
         $j \leftarrow -1$ 
      else
        intercambiar( $a[j]$ ,  $a[k]$ )
         $j \leftarrow j - \text{intervalo}$ 
      end if
    end while
  end for
end while
  
```

Pregunta 3 (2 puntos) ¿Se puede afirmar que la búsqueda binaria es siempre la mejor opción para realizar una búsqueda en un conjunto de datos? **Justifique adecuadamente su respuesta.**

PARTE PRÁCTICA

Pregunta 4 (5 puntos) Las listas invertidas son estructuras de datos usadas en los sistemas de recuperación de información. Estas sirven como estructura auxiliar para la recuperación de documentos que contienen determinados términos de búsqueda.

Se desea simular la creación de una lista invertida usando archivos binarios. Para esto se deberán leer todas las palabras de un conjunto grande de archivos de texto y tratar cada palabra como si fuera un término. Para cada término de la lista se deberá guardar el archivo de texto que la contiene y la frecuencia del término en el archivo de texto.

Dado el archivo de texto `dir.txt` que contiene en cada línea el nombre de un archivo de texto, se pide que simule la creación de un vector invertido usando archivos binarios. El nombre de cada archivo binario será el nombre de cada término obtenido de cada archivo de texto. La **stop list** estará dada por un archivo de texto, el cual será el primero listado en `dir.txt`. En el archivo de la **stop list** se tendrá una palabra por línea. Los términos que coincidan con las palabras de la **stop list** no deberán ser considerados en la lista invertida.

Asuma que en los archivos existen palabras sin signos de puntuación.

- a) (1.5 puntos) Describa la estrategia que usará para la creación de la lista invertida
- b) (3.5 puntos) Implemente la lista invertida usando ANSI C.

Pregunta 5 (4 puntos)

- a) (2.5 puntos) El **problema de la bandera nacional de Holanda** consiste en reordenar un arreglo que contiene caracteres R, B y A (rojo, blanco y azul son los colores de la bandera nacional holandesa) de tal forma que todas las R's aparezcan primero, las B's vengan después, y las A's estén al final. Diseñe un algoritmo **lineal in-place** (i.e., que ordene los elementos en el mismo arreglo y no en una estructura auxiliar) para resolver este problema.
- b) (1.5 puntos) Explique cómo la solución al problema de la bandera nacional de Holanda puede ser usada en el algoritmo **quicksort**.

PARTE ELECTIVA

Para **UNA** de las preguntas que se presentan a continuación, elabore un programa en ANSI C que resuelva el problema descrito.

Pregunta 6 (6 puntos) UVa 10422 - Knights in FEN (Traducción Libre) Se tienen caballos blancos y negros en un tablero de ajedrez de 5x5. Existen 12 caballos de cada color, además de un casillero que está vacío. En cualquier momento, un caballo puede moverse a un casillero vacío, siempre y cuando se mueva como un caballo en un juego normal de ajedrez (¿qué más esperaba?).

Dada una configuración inicial del tablero, la pregunta es: ¿cuál es el mínimo número de movimientos en los que se puede alcanzar la configuración final (Fig. 1)?

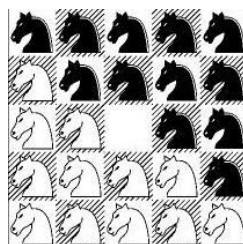


Figura 1: Configuración **final** de los caballos en el tablero de ajedrez.

Entrada: La primera línea de la entrada contiene un número entero N ($N < 14$) que indica cuántos conjuntos de prueba existen. Cada conjunto consiste de cinco líneas, en la que cada una representa una fila de un tablero de ajedrez. Las posiciones ocupadas por los caballos blancos están marcadas como 0 y las posiciones ocupadas por los caballos negros están marcadas como 1. El espacio corresponde al casillero vacío en el tablero. No existen líneas en blanco entre conjuntos de prueba.

Por ejemplo, el primer conjunto de prueba del ejemplo de entrada corresponde a la siguiente configuración inicial:

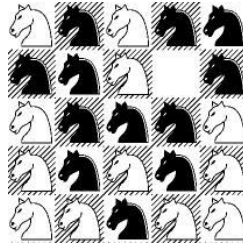


Figura 2: Ejemplo de configuración **inicial** de los caballos en el tablero de ajedrez.

Salida: Para cada conjunto de prueba, su tarea es encontrar el mínimo número de movimientos que van desde la configuración inicial especificada hasta la configuración final (Fig. 1). Si ese número es mayor que 10 entonces la salida debe ser una línea con el texto:

No se puede resolver en menos de 11 movimientos.

En caso contrario, la salida debe ser:

Se puede resolver en n movimiento(s).

donde $n \leq 10$. La salida para cada conjunto de prueba se debe presentar en una única línea, como se muestra en el ejemplo.

Ejemplo de Entrada:

```
2
01011
110 1
01110
01010
00100
10110
01 11
10111
01001
00000
```

Ejemplo de Salida:

```
No se puede resolver en menos de 11 movimientos.
Se puede resolver en 7 movimiento(s).
```

Pregunta 7 (6 puntos) UVa 990 - Diving for Gold (Traducción Libre) A John le gusta bucear y cazar tesoros. Para su buena suerte, él acaba de encontrar la ubicación de un barco pirata lleno de tesoros. El sofisticado sistema sonar a bordo de su barco le permite identificar la ubicación, profundidad y cantidad de oro en cada tesoro hundido. Desafortunadamente, John olvidó traer un dispositivo GPS y las posibilidades de encontrar esta ubicación de nuevo son muy escasas, por lo que tiene que apropiarse del oro ahora. Para empeorar la situación, John solo tiene un tanque de aire comprimido.

John quiere bucear con el tanque de aire comprimido para recuperar tanto oro como sea posible del naufragio. Escriba un programa que John pueda usar para seleccionar cuáles tesoros debe recoger para maximizar la cantidad de oro recuperado.

El problema tiene las siguientes restricciones:

- Existen n tesoros $\{(d_1, v_1), (d_2, v_2), \dots, (d_n, v_n)\}$ representados por el par (profundidad, cantidad de oro). Existen como máximo 30 tesoros.
- El tanque de aire solo permite estar t segundos bajo el agua. t es como máximo 1000 segundos.
- En cada inmersión, John puede traer como máximo un tesoro a la vez.
- El tiempo para descender es $td_i = w * d_i$, donde w es un número entero constante.
- El tiempo para ascender es $ta_i = 2w * d_i$, donde w es un número entero constante.
- Debido a limitaciones de los instrumentos, todos los parámetros son números enteros.

Entrada: La entrada para este programa consiste de una secuencia de valores enteros. La entrada contiene varios casos de prueba. La primera línea de cada conjunto de prueba contiene los valores t y w . La segunda línea contiene el número de tesoros. Cada una de las siguientes líneas contiene la profundidad d_i y la cantidad de oro v_i de un tesoro diferente. Una línea en blanco separa cada caso de prueba.

Salida: La primera línea de la salida para cada conjunto de prueba debe contener la máxima cantidad de oro que John puede recoger del naufragio. La segunda línea debe contener el número de tesoros recuperados. Cada una de las siguientes líneas debe contener la profundidad y la cantidad de oro de cada tesoro recuperado. Los tesoros deben ser presentados en el mismo orden que en la entrada. Imprima una línea en blanco entre las salidas de diferentes conjuntos de prueba.

Ejemplo de Entrada:

```
210 4
3
10 5
10 1
7 2
```

En este ejemplo de entrada, el tanque de aire comprimido tiene una capacidad de 210 segundos, la constante w tiene el valor de 4 y existen 3 tesoros. El primero está a una profundidad de 10 metros y vale 5 monedas de oro, el segundo está a una profundidad de 10 metros y vale 1 moneda de oro, y el tercero está a una profundidad de 7 metros y vale 2 monedas de oro.

Ejemplo de Salida:

```
7
2
10 5
7 2
```

Profesores del curso: Andrés Melgar
 Fernando Alva

Pando, 19 de octubre de 2013