

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Segundo Examen

(Primer Semestre 2022)

Duración: 2h 50 min.

- Al inicio de cada programa, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- Los programas deben ser desarrollados en Ansi C. **Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.**
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- El orden será parte de la evaluación.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- **Solo puede usar código brindado en clases o subido a PAIDEIA, si emplea código de otras fuentes se considera plagio.**
- Para este examen solo se permite el uso de las librerías `stdio.h`, `stdlib.h`, `string.h` y `math.h`
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_EX2_PY`
- Su trabajo deberá ser subido a PAIDEIA.

Pregunta 1 (10 puntos)

La empresa ROBO.T.ROBO, en la cual usted trabaja, ha comprado un robot de dudosa procedencia. Los anuncios especifican que dicho robot es capaz de resolver criptogramas matemáticos (operaciones matemáticas en las que los números han sido sustituidos por otros caracteres, y se deben encontrar dichos valores). Su jefe no está seguro de que el robot sea capaz de cumplir con su objetivo, por lo que le ha solicitado construir un algoritmo para descifrar el criptograma, y luego compararlo con los resultados del robot. Un ejemplo de criptograma matemático podría ser el siguiente:

$$\begin{array}{ccccccc} & D & O & S & + & & \\ & D & O & S & & & \\ & D & O & S & & & \\ \hline & D & O & S & & & \\ & O & C & H & O & & \end{array}$$

Que tiene como algunas de sus respuestas a las siguientes combinaciones:

D = 7, O = 2, S = 8, C = 9, H = 1

D = 6, O = 2, S = 3, C = 4, H = 9

D = 7, O = 2, S = 3, C = 8, H = 9

D = 5, O = 2, S = 3, C = 0, H = 9

D = 6, O = 2, S = 8, C = 5, H = 1

Tenga en cuenta que, para este caso, la letra O no podría ser 0, ya que ningún número real debería empezar por ceros, y que dos letras no pueden representar al mismo número.

Se solicita implementar lo siguiente:

- Desarrollar las estructuras y/o funciones necesarias para el ingreso de datos. Puede ser por teclado o por archivo (adjuntar el archivo con su entrega). (1.0 punto)
- Desarrollar un algoritmo que utilice **backtracking** para obtener todas las posibles soluciones a cualquier criptograma propuestos. En caso no exista solución, indicarlo al usuario. (9.0 puntos)

Puede considerar siempre la misma cantidad filas similar al ejemplo.

Pregunta 2 (10 puntos)

Una empresa productora de artefactos ensamblados desea adquirir más de un robot para realizar diferentes tareas ya que, si compra uno solo, todas las tareas deberían realizarse de forma secuencial, por tal motivo el tiempo de desarrollo de un producto sería muy largo. Desde luego tampoco se cuenta con el dinero para comprar robots en exceso que no tengan una labor asignada. Por tal motivo los gerentes de producción desean saber cuanto tiempo como máximo se demorarían **R** robots en realizar un conjunto de **N** tareas, donde cada una tiene tiempos que pueden ser diferentes o no. A continuación, se muestra un ejemplo:

Datos de Ingreso: N = 5

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5
Tiempo (min)	15	30	60	45	10

Para R = 2

El tiempo máximo que utilizarían 2 robots sería 100 min.

El detalle de este resultado sería un robot realizará la tarea 3 y el otro realizará las tareas 1, 2, 4 y 5

Para R = 3

El tiempo máximo que utilizarían 3 robots sería 60 min.

El detalle de este resultado sería un robot realizará la tarea 3, otro realizará las tareas 1, 2 y 5, el tercer robot se encargará de la tarea 4.

Se solicita:

- Utilizando **programación dinámica**, desarrolle un programa que calcule el tiempo máximo de demora para un grupo de tareas con un determinado número de robots. Recuerde que **solo** debe mostrar el tiempo que demora, no es necesario mostrar el detalle de la asignación. Para esta pregunta debe usar los datos que se muestran en el ejemplo, **no** deben ingresarlos por el teclado o archivo (5.0 puntos)

Una casa de cambio analiza que las variaciones del tipo de cambio entre soles y dólares son bastante usuales, lo cual representa una oportunidad para adquirir dólares cuando la moneda tenga un precio bajo y venderlos cuando el precio suba. Para culminar su análisis los cambistas desean estimar cuanto hubieran ganado al realizar **K** compras en un determinado rango de **N** fechas. A continuación, se muestra un ejemplo:

Datos de Ingreso: N = 6

T/C	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6
Para 100\$	315	322	385	375	365	380

Para K = 1 (una compra de dólares)

La ganancia máxima será 70 soles para \$100

El detalle de este resultado sería comprar el día 1 y vender el día 3.

Para K = 2 (dos compras de dólares)

La ganancia máxima será 85 soles para \$100

El detalle de este resultado sería una compra el día 1 y vender el día 3, luego realizar una segunda compra el día 5 y vende el día 6.

Para resolver el problema se le solicita:

- b) Utilizando **programación dinámica**, desarrolle un programe que le ayude a resolver este problema, que es calcular la máxima ganancia de acuerdo con la cantidad de compras de dólares. Recuerde que solo debe mostrar la ganancia, no es necesario mostrar el detalle de la asignación. Para esta pregunta debe usar los datos que se muestran en el ejemplo, **no** deben ingresarlos por el teclado o archivo. No debe reordenar los valores del tipo de cambio, ya que los días determinan las variaciones en el precio (5.0 puntos)

Para ambas preguntas debe mostrar la matriz de soluciones propia de la programación dinámica. Además, debe presentar dos proyectos o archivos .c de acuerdo con la opción desarrollada.

Pregunta 3 (10 puntos)

BALWORD es un juego en el que se elige una palabra de n letras desde un archivo de texto y el jugador intenta adivinar dentro de n intentos. Después de que el jugador introduce la palabra que supone que es, cada letra se muestra si es que es correcta y está en la posición correcta, la letra que está en la palabra incógnita pero no en la posición correcta se muestra con una marca (x) al costado de la letra, y la letra que no está en la palabra incógnita muestra el símbolo '_'. El desarrollo del juego deberá utilizar sólo un TAD de tipo árbol y recursividad en todo momento, tanto para la lectura de la palabra incógnita como para el desarrollo de la solución.

Por ejemplo, si la palabra incógnita tiene 5 letras y es "HELIO", dos simulaciones del juego son:

```
BIENVENIDO AL JUEGO DE BALWORD
Descubre la palabra de 5 letras. Tiene máximo
5 intentos.

Intento 1: ADORA
  _ _ O(x) _ _
Intento 2: TEORIA
La longitud de la palabra ingresada no es la
correcta. Vuelva a intentarlo.
Intento 2: OMITE
O(x) _ I(x) _ E(x)
Intento 3: SERIO
  _ E _ I O
Intento 4: TEDIO
  _ E _ I O
Intento 5: CESIO
  _ E _ I O
Lo siento, se agotaron los intentos.
```

```
BIENVENIDO AL JUEGO DE BALWORD
Descubre la palabra de 5 letras. Tiene máximo
5 intentos.

Intento 1: GENIO
  _ E _ I O
Intento 2: HELIO
H E L I O

Felicitaciones por haber resuelto el reto.
```

Para la lectura de la palabra incógnita se realizará mediante un archivo de texto en el que el primer dato es la cantidad de letras y por cada letra un índice de la letra que indica su posición en la palabra y por supuesto, la letra misma. Las letras de la palabra incógnita se guardarán en un ABB. Por ejemplo, si el archivo contiene los datos de la Tabla 1, entonces el ABB formado será el presentado en la Figura 2.

Tabla 1. Ejemplo de archivo con los datos de entrada

5	
2	E
4	I
5	O
3	L
1	H

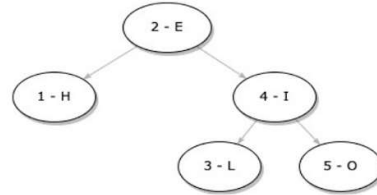


Figura 1. ABB de letras de la palabra incógnita.

Se le solicita implementar lo siguiente:

- Desarrolle las estructuras y funciones necesarias para el ingreso de los datos en el ABB de letras. La lectura de datos debe utilizar una estrategia recursiva y la cantidad de letras es variable. (2.0 puntos)
- Desarrolle una o varias funciones que permitan determinar si una palabra coincide o no con la palabra incógnita del ABB de letras. Para el desarrollo **NO puede utilizar otra TAD ni arreglos**, sólo el ABB de letras. Puede añadir los atributos que considere necesarios en los nodos del ABB. Recuerde que sólo puede utilizar estrategias recursivas. (5.0 puntos)
- Desarrolle la función principal que, luego del ingreso de letras al ABB desde el archivo, muestre la bienvenida al jugador, muestre el símbolo ' _ ' por cada letra de la palabra incógnita, y por cada intento, solicite la palabra al jugador, valide la longitud de la palabra ingresada, si la longitud de la palabra ingresada es válida, se procede a determinar si la palabra coincide con la palabra incógnita, muestre las letras de la palabra de acuerdo con la regla descrita en el primer párrafo del enunciado. Si no se descubre la palabra luego de los n intentos, se muestra el mensaje que está al final de la simulación 1. Si se descubre la palabra dentro de los n intentos, se muestra el mensaje que está al final de la simulación 2. Para el desarrollo sólo puede utilizar **recursividad** y **NO puede utilizar otra TAD ni arreglos**, sólo el ABB de letras. (3.0 puntos)

Profesores del curso:

Johan Baldeon
Rony Cueva
Juan Manuel Chau

San Miguel, 9 de julio del 2022