

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Segundo Examen

(Primer Semestre 2021)

Duración: 2h 50 min.

- Debe **elegir y responder sólo 2 preguntas** de las 3 propuestas. Las dos preguntas elegidas deben sumar 20 puntos.
- En cada función el alumno deberá incluir, a modo de comentario, la estrategia o forma de solución que utiliza 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 el lenguaje 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.
- Para este examen solo se permite el uso de las librerías **stdio.h**, **stdlib.h**, **math.h** y **string.h**
- Su trabajo deberá ser subido a PAIDEIA.
- Los archivos deben llevar como nombre su código de la siguiente forma **codigo_EX2_PY**

Pregunta 1 (10 puntos)

En todo el país el proceso de vacunación está avanzando muy rápidamente, por tal motivo el Ministerio de Salud a cargo ha decidido adquirir 4 tipos de vacunas diferentes: Fisher, Spamtik, Shinovac y Astraazteca, cada una tiene un costo diferente de implementación en una ciudad. Además, la vacuna Fisher necesita que la ciudad cuente con equipo de refrigeración especial, en caso contrario, no se puede utilizar. Adicionalmente, considerando que no se debe usar una sola vacuna en todas las ciudades, el Ministerio de Salud ha decidido que 2 ciudades colindantes siempre utilicen vacunas diferentes. Por ejemplo, si consideramos 6 ciudades:

Ciudad	Ciudades Colindantes	Cuenta con equipo de refrigeración especial
0	1,2,5	No
1	0,2,3	No
2	0,1,3,5	Sí
3	1,2,4,5	No
4	3,5	Sí
5	0,2,3,4	No

El costo de implementar cada tipo de vacuna en una ciudad es el siguiente:

Fisher \$ 10K, Spamtik \$ 5K , Shinovac \$ 12K y Astraazteca \$8K.

Por ejemplo, una solución válida para este conjunto de datos, si consideramos Fisher (1), Spamtik (2), Shinovac (3) y Astraazteca (4) sería:

Ciudad	0	1	2	3	4	5
Vacuna	2	4	1	2	1	3

Con un costo de \$ 50 K

Se solicita que desarrolle un programa que, utilizando **backtracking**, ayude a encontrar las soluciones a este problema. Para probar su programa puede utilizar los datos que se muestran en el ejemplo o ingresarlos por teclado o archivo de texto.

- Implemente una **función** que, utilizando **backtracking**, muestre todas las posibles soluciones de distribución de las vacunas por cada ciudad. Considere los argumentos que cree necesarios para la función. Esta función debe ser invocada en el programa principal **(5.0 puntos)**.
- Realice las modificaciones necesarias en la función realizada en la parte (a), y manteniendo su estrategia, para calcular el costo de cada una de las distribuciones de las vacunas en las ciudades. No puede realizar este cálculo fuera de la función **(5.0 puntos)**.

Pregunta 2 (10 puntos)

Como agente corredor de bolsa, es el responsable de la compra de acciones de la empresa JBM y por día sólo puede comprar una cantidad límite de **T** acciones. El precio de una acción para ese día es de **P** dólares. **N** clientes están interesados en comprar acciones de la empresa, y cada uno de ellos está interesado en comprar **S_i** acciones, donde $0 < i \leq N$. A su vez, cada cliente está dispuesto a pagarle una comisión que se obtiene por un porcentaje **C_i** del monto por la compra de las acciones para el cliente *i*, donde $0 < i \leq N$. El porcentaje **C_i** de comisión es variable por cada cliente.

Usted debe elegir a los clientes que le brinden la mayor ganancia gracias a la suma de los montos por las comisiones de las compras de las acciones por cada cliente. Recuerde que la suma de todas las acciones que puede comprar para los clientes elegidos no debe de exceder la cantidad de **T** acciones.

Cada cliente tiene los siguientes datos: id, nombre, cantidad de acciones a comprar, porcentaje de comisión y ganancia que le daría gracias a la comisión aplicada al monto por la compra de acciones. El id es entero, el nombre es alfanumérico de máximo 50 caracteres, la cantidad de acciones a comprar es entera, el porcentaje de comisión es un número con coma decimal y la ganancia sería real.

A continuación, se muestra un ejemplo:

Datos de entrada:

T = 10000, **P** = 4, **N** = 6

Id Cliente	1	2	3	4	5	6
Nombre	John	Ron	Micky	Vicky	Ceci	Isa
Cant. Acciones (S _i)	3000	3800	2800	3500	3200	3400
% de comisión (C _i)	0.1	0.02	0.15	0.05	0.08	0.03

Resultado:

Para este conjunto de datos, la máxima ganancia debido a las comisiones sería de **3904.00** dólares por un total de **9000** acciones.

Clientes que se deben de elegir para obtener la mayor ganancia:

- Cliente **1** - **John**, ganancia = **1200.00** dólares por comprar **3000** acciones.
- Cliente **3** - **Micky**, ganancia = **1680.00** dólares por comprar **2800** acciones.
- Cliente **5** - **Ceci**, ganancia = **1024.00** dólares por comprar **3200** acciones.

- a) Desarrolle las estructuras e instrucciones necesarias para el ingreso de datos al programa. Los datos de entrada pueden ser ingresados por teclado o por archivo **(1.0 punto)**.
- b) Implemente una función que, utilizando **programación dinámica**, calcule la máxima ganancia en dólares por la suma de montos por comisiones de la compra de acciones de un conjunto de clientes de acuerdo con las restricciones del enunciado, y la cantidad de acciones en total que se pueden comprar. Considere los argumentos que cree necesarios para la función. Esta función debe ser invocada en el programa principal **(6.0 puntos)**.
- c) Realice las modificaciones necesarias en su programa para que, de acuerdo con el proceso realizado en la parte (b), y manteniendo la **programación dinámica**, se pueda identificar los clientes que se deben de elegir para obtener la máxima ganancia, así como mostrar las ganancias en dólares y la cantidad de acciones por cada cliente. Si en caso hay más de una solución puede mostrar solo una de ellas **(3.0 puntos)**.

Pregunta 3 (10 puntos)

a) **(5.0 puntos)** Desarrolle un programa utilizando **backtracking** que, dada una cadena **s** que contiene solo dígitos, devuelva todas las posibles direcciones IP válidas que se pueden obtener de **s**. Puede devolverlos en cualquier orden. La longitud de la cadena **s** debe ser máxima de 12 caracteres.

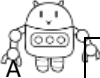
Una dirección IP válida consta de exactamente cuatro números enteros, cada número entero está entre 0 y 255, separados por puntos simples y no pueden tener ceros a la izquierda. Por ejemplo, "0.1.2.201" y "192.168.1.1" son direcciones IP válidas y "0.011.255.245", "192.168.1.312" y "192.168.1.1.1" son direcciones IP no válidas.

A continuación, se presentan algunos ejemplos de ejecución:

Ejemplo	Entrada	Resultado
1	25525511169	255.255.11.169 255.255.111.69
2	0000	0.0.0.0
3	1111	1.1.1.1
4	010010	0.10.0.10 0.100.1.0
5	101069	1.0.10.69 1.0.106.9 10.10.6.9 10.1.0.69 101.0.6.9
6	669255255255	(No presenta resultados)

7	0011255245	(No presenta resultados)
8	00001	(No presenta resultados)

b) (5.0 puntos) Una empresa dedicada a la venta de productos alimenticios, ha adquirido un robot para que pueda ingresar al congelador y sacar los productos que se encuentran almacenados en esa área. Los productos de este congelador son artículos cárnicos y están embolsados por peso, se sabe que este congelador está dividido en 2 bloques (izquierda y derecha), cada bloque tiene 2 pisos (A y B) y además tiene varias secciones que van de 1 a **N**. Usualmente el robot recibe un pedido que está representado por el peso total que debe sacar del congelador, así que debe ingresar a esta área y juntar los productos de acuerdo con el peso solicitado. A continuación, un ejemplo de almacén de $N=4$ secciones con la distribución de los artículos cárnicos, en el que cada celda se encuentra el peso de su correspondiente artículo:

		1	2	3	4
IZQUIERDA	A	7	17	14	19
	B	9	12	11	8
					
DERECHA	A	8	6	10	12
	B	18	7	15	11

Si se le solicita al robot formar un pedido de 23 kilos, podrá obtener las siguientes soluciones:

Resultado 1: 7k 9k 7k Ubicaciones: IA1 IB1 DB2

Resultado 2: 9k 14k Ubicaciones: IB1 IA3

Para realizar esta tarea se le solicita que desarrolle un programa que, usando **backtracking** o **programación dinámica**, muestre los pesos de los productos seleccionados y sus ubicaciones, como indica el ejemplo, de acuerdo con el peso total del pedido. Si usa backtracking debe obtener todos los resultados posibles; si usa programación dinámica, solo tiene que buscar un resultado. Puede usar los datos mostrados en el ejemplo, para evitar ingreso de datos.

Profesores del curso:

Johan Baldeón
Rony Cueva

San Miguel, 24 de julio del 2021