

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Segundo Examen

(Segundo Semestre 2020)

Duración: 2h 50 min.

- 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 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.
- Para este examen solo se permite el uso de las librerías **stdio.h**, **stdlib.h** y **math.h**
- **La interpretación de las preguntas es parte de la evaluación.**
- Su trabajo deberá ser subido a PAIDEIA.
- Los archivos deben llevar como nombre su código de la siguiente forma **codigo_EX2_PY**

Escoja 2 preguntas de las 4 propuestas:

Pregunta 1 (10 puntos)

Una empresa dedicada a la comercialización de productos ferreteros, cuenta con racks o ubicaciones de almacenamiento, para colocar los productos que vende. Por tal motivo ha adquirido un robot que ayude a las operaciones de ingreso o almacenamiento de productos en el almacén. Para que trabaje correctamente se le solicita a Ud. como especialista, que desarrolle un algoritmo basado en **backtracking** para el robot adquirido. El autómata debe ser capaz de colocar los productos recibidos en los espacios disponibles y mostrar las cantidades ubicadas luego del proceso. Los productos recibidos vienen agrupados por cantidades, por lo cual no pueden dividirse, solo colocarse en las ubicaciones que pueden contenerlos. Una ubicación puede almacenar más de un grupo de productos de un mismo ingreso. Los ingresos de productos a almacenar tienen un tamaño N y la cantidad de ubicaciones del rack es de tamaño M. Se puede cambiar el orden de los productos a ingresar, pero el orden de las ubicaciones disponibles es fijo. Por ejemplo:

Para un **ingreso** de $N = 4$, con productos agrupados con cantidades $\{2,3,1,4\}$. Se cuenta con un rack de $M = 5$ ubicaciones, el cual tiene las siguientes capacidades disponibles $\{3,2,5,1,3\}$

Los resultados de ubicar los productos serán:

$\{3, 0, 4, 0, 3\}$ $\{3, 2, 5, 0, 0\}$ $\{3, 1, 4, 0, 2\}$

La primera respuesta se interpreta: se almaceno 3 productos en la ubicación 0, se almaceno 4 productos en la ubicación 2, se almaceno 3 productos en la ubicación 4 (2 y 1 productos agrupados).

- Desarrolle las funciones necesarias para el soporte de esta solicitud, así como el ingreso de datos, vía teclado o archivo de texto. Los datos por ingresar son: N, M, las cantidades de cada ingreso y las capacidades de cada ubicación del rack (1 punto)
- Desarrolle una función que, utilizando la estrategia mencionada anteriormente, imprima todas las posibles respuestas de ubicación de los productos, según el ejemplo mostrado anteriormente (6.0 puntos)
- Modifique el algoritmo para que ahora la función, imprima todas las posibles combinaciones de ingreso de productos, de acuerdo con el orden de almacenamiento en el rack (3.0 puntos).

Pregunta 2 (10 puntos)

Una empresa de producción desea emplear un ABB para el control de su plan de producción mensual. Se sabe que, por temas mecánicos, siempre se le da **prioridad** a las ordenes de **mayor** cantidad de productos. Además, las ordenes se atienden por semana y la empresa tiene una capacidad mensual de producción P, donde $P \leq 100,000$ productos, esta cantidad se distribuye uniformemente entre las 4 semanas. La fábrica mensualmente programa N productos diferentes y $N < 10$. Para construir el árbol se consideran algunas reglas:

- Los nodos deben contener el id del producto, semana y cantidad de producir. Pueden tener menos campos si lo desean.
- Las ordenes siempre se registran en el árbol de la siguiente forma, primero las de tercera semana, luego cuarta, primera y segunda.
- En el árbol las ordenes deben estar ordenadas primero por la semana y luego por la cantidad, de acuerdo con la prioridad establecida.
- Si en caso en una semana hay 2 ordenes con la misma cantidad, el id del producto determina cual es menor o mayor.

El **proceso** de atención de ordenes se realiza bajo las siguientes reglas:

- De acuerdo con la capacidad mensual de la fábrica, se determina la capacidad semanal.
- Utilizando el orden del ABB, se deben imprimir las ordenes que pueden ser atendidas por la capacidad semanal (muestre a que semana corresponden). Estas órdenes deben actualizarse de acuerdo con la cantidad atendida.
- Luego se debe imprimir en preorden como queda el árbol.
- Las ordenes no atendidas se deben colocar en una lista de nodos, retirándose del árbol.
- Luego se debe repetir el proceso para la semana siguiente.

Por ejemplo, los datos recibidos pueden ser de la siguiente forma:

$P = 40$ $K = 5$. Las cantidades semanales se expresan en Miles de productos(K).

Producto	Semana 1	Semana 2	Semana 3	Semana 4
1	4	0	2	0
2	2	0	3	3
3	8	0	4	2
4	0	3	0	0
5	2	2	0	0

- Diseñe las estructuras válidas para soportar el programa solicitado, recuerde que los datos a ingresar son N, P y la tabla de productos con las cantidades de las órdenes. Debe utilizar para el ingreso el teclado o archivos de texto. Imprima el árbol cargado en preorden (1.0 punto).
- Desarrolle un programa que realice el **proceso** de atención de ordenes solicitadas para todo el mes. Las funciones de búsqueda (buscar elemento, buscar mínimo, etc.) deben ser obligatoriamente **iterativas** (6.0 puntos).

- c) Desarrolle una función de recorrido **inorden iterativa** para imprimir todo árbol antes del proceso de atención. Si no se realiza la pregunta b, esta pregunta no tiene valor. (3.0 puntos).

No puede usar estructuras auxiliares, con excepción de las funciones iterativas donde puede usar TAD's.

Pregunta 3 (10 puntos)

Luego de la campaña navideña, una empresa de productos avícolas desea regalar los últimos productos que le quedan dentro de su almacén de tamaño $n \times n$, donde en cada ubicación se almacena un determinado producto identificado por su peso. Por tal motivo la gerencia desea que Ud. desarrolle una función que utilice **programación dinámica** para resolver los diferentes escenarios planteados. Por ejemplo:

Ingreso de datos: $n = 5$, con los siguientes productos en el almacén:

0	0	6	0	0
0	0	0	0	0
0	0	4	0	0
0	0	0	0	0
3	2	0	0	3

- a) La gerencia desea repartir todos los productos que le quedan entre sus 3 mejores clientes, por tal motivo desarrolle una función basada en la estrategia requerida, que verifique si los productos se pueden repartir equitativamente entre los clientes de acuerdo con el peso, si es viable imprima una solución de esta división (5 puntos)
- b) La gerencia no ha quedado satisfecha con regalar los productos a sus 3 mejores clientes, por tal motivo ahora desea saber de cuántas formas es posible agruparlos para regalarlo a sus mejores trabajadores, considerando que cada producto se puede regalar por separado o se puede juntar a otro en pares, desarrolle una función que, utilizando la estrategia indicada anteriormente, le permita calcular la cantidad de formas en que puede regalar los productos. Para el ejemplo mostrado se puede agrupar de 26 formas posibles (5 puntos).

Pregunta 4 (10 puntos)

Un horno de producción de galletas tiene 10 secciones enlazadas consecutivas. Anualmente se le realiza un mantenimiento preventivo, para revisar si alguna de estas secciones tiene fallas. Si alguna de las mismas presenta desperfectos se tienen que importar desde Europa la nueva sección, lo que origina que su traslado y ensamble sea muy costoso, por lo cual se prefiere siempre comprar las secciones por grupos. Los grupos pueden estar formados por una o más secciones consecutivas. Cuando se realiza la revisión y se detecta que mas de 5 grupos de secciones presentan fallas, se prefiere reemplazar todo el horno. Las secciones y grupos de secciones se representan de la siguiente forma, por ejemplo:

Horno completo, los ceros representan las secciones con fallas:

(1 - 1) -> (2 - 1) -> (3 - 1) -> (4 - 0) -> (5 - 0) -> (6 - 1) -> (7 - 0) -> (8 - 0) -> (9 - 1) -> (10 - 0)

Como se aprecia se han detectado fallas en las secciones: 4,5,7,8 y 10.

Por tal motivo se deben reemplazar 3 grupos de secciones, por nuevas, como las siguientes:

Grupo 1: (4 - 1) -> (5 - 1)

Grupo 2: (7 - 1) -> (8 - 1)

Grupo 3: (10 - 1)

Ya que el costo de ensamble es muy alto, y el ensamblado de fábrica es muy bueno, se prefiere reemplazar las secciones sin separar los grupos de secciones importadas. Por tal motivo se le solicita a Ud. que utilizando listas enlazadas (debe definir una lista enlazada para el horno a reparar y listas para cada grupo de secciones nuevas) realice las siguientes tareas:

- a) Diseñe las estructuras y funciones válidas para el soporte al programa solicitado, debe ingresar las secciones defectuosas de la línea y los 5 posibles grupos a reemplazar si es necesario. Debe utilizar para el ingreso el teclado o archivos de texto. Finalmente imprima la lista de forma similar al ejemplo (1.0 punto).
- b) Desarrolle una función, que reemplace los grupos de secciones defectuosos. Para realizar esta tarea no puede dividir los grupos de secciones nuevas, además solo puede recorrer una sola vez la lista que representa al horno. No puede usar ninguna estructura auxiliar o crear memoria (malloc) para esta operación, solo punteros. Finalmente imprima la lista que representa al horno reparado (9 puntos).

Profesor del curso:

Rony Cueva

San Miguel, 19 de diciembre del 2020