

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMOS AVANZADOS

**Primer Examen
(Primer Semestre 2025)**

Duración: 2h 50 min.

- **No puede utilizar apuntes, solo hojas sueltas en blanco.**
- En cada función el alumno deberá incluir, a modo de comentario, la forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear plantillas o funciones no vistas en los cursos de programación de la especialidad.
- 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.
- **Solo está permitido acceder a la plataforma de PAIDEIA, cualquier tipo de navegación, búsqueda o uso de herramientas de comunicación se considera plagio por tal motivo se anulará la evaluación y se procederá con las medidas disciplinarias dispuestas por la FCI.**
- Para esta evaluación solo se permite el uso de las librerías **iostream, iomanip, climits cmath, fstream, vector, algorithm, string o cstring**
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans.**
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_EX1_P#` (donde # representa el número de la pregunta a resolver)

Pregunta 1 (10 puntos)

La empresa en la que trabajas, FutureDelivery, se dedica a realizar entregas de productos por medio de drones. El uso de este medio facilita el diseño de rutas basadas en viajes directos (líneas rectas) entre locaciones definidas por coordenadas geográficas.

FutureDelivery te ha contratado para diseñar e implementar algoritmos que seleccionen rutas eficientes para la entrega de productos. A cada drone se le asigna una orden definida por un grupo de puntos (pedidos) representados por pares de coordenadas. Por ejemplo, una orden de 5 pedidos puede ser la siguiente:

	0	1	2	3	4	5
Coordenadas	(2,3)	(1,1)	(1,5.5)	(6,3)	(2.5,1)	(3.5, 6)

Donde el punto 0 es el almacén central, con coordenadas (2,3). El primer paso para el diseño de rutas es el cálculo de la matriz de distancias entre todos los puntos de la orden. Por ejemplo, los 6 puntos de la orden de arriba genera la siguiente matriz de distancias:

	0	1	2	3	4	5
0	0.00	2.24	2.69	4.00	2.06	3.35
1	2.24	0.00	4.50	5.39	1.50	5.59

2	2.69	4.50	0.00	5.59	4.74	2.55
3	4.00	5.39	5.59	0.00	4.03	3.91
4	2.06	1.50	4.74	4.03	0.00	5.10
5	3.35	5.59	2.55	3.91	5.10	0.00

Para cada orden se nos pide diseñar la ruta con menor distancia total, con la condición de que el dron debe de partir del almacén central (punto 0), visitar cada punto una sola vez, y regresar al almacén de donde partió. Es decir, las rutas son ciclos que empiezan y acaban en el almacén central.

Decidimos atacar el problema primero generando todas las posibles rutas de la orden (brute force) y escogiendo la ruta (ciclo) de menor distancia. Sabemos que esto no es eficiente, pero al menos así nos aseguramos de generar una solución correcta que podemos usar luego para verificar futuras optimizaciones. Notamos que, dadas las condiciones del problema y a diferencia de otros problemas de rutas, el dron puede potencialmente viajar entre cualquier par de puntos de la orden. Esto nos hace acordar al problema de colocar n-torres en un tablero de ajedrez.

Tarea: Dada la matriz de distancias correspondiente a una orden, implementar un algoritmo usando backtracking que genere **todas** las posibles rutas validas y calcule la ruta optima, así como el número de rutas generadas.

Por ejemplo, dada la matriz de distancias correspondiente a la orden del ejemplo de arriba, el algoritmo imprimirá:

Número de rutas generadas: 120

Ruta optima: 0 1 4 3 5 2 (distancia 16.92)

Donde el último punto visitado en la ruta antes de regresar al almacén es 2. La distancia total incluye la distancia de 2 a 0.

Sugerencia: Diseñar el algoritmo de tal manera que el cálculo de mínimos se realiza al final de la construcción de cada ruta/ciclo, comparando el valor del ciclo generado con el valor del mínimo calculado hasta ese momento (mínimo actual).

Pregunta 2 (10 puntos)

Una empresa de almacenes esta tratando de armar pilas de paquetes paralelepípedos lo más alto posible, estos paquetes tienen diversas dimensiones y pesos, para este proceso se cuenta con las siguientes restricciones y detalles:

- No se puede apilar un paquete de menor dimensión bajo otro de mayor dimensión, no importa que su área sea mayor. Por ejemplo, no se puede colocar una caja de 1x10 debajo de una caja de 2x2.
- Existen varias cajas con las mismas dimensiones por tal motivo se pueden considerar para la búsqueda de la solución.
- Las cajas se pueden rotar
- No se puede colocar una caja con un peso mayor encima de otra, con el mismo peso si es posible.

A continuación, algunos ejemplos:

Medidas	Paquete 1	Paquete 2
Lado 1	1	2
Lado 2	3	15
Lado 3	4	6

Peso	15	10
------	----	----

Respuesta: La altura máxima es 17

Medidas	Paquete 1	Paquete 2
Lado 1	1	2
Lado 2	3	15
Lado 3	4	6
Peso	2	10

Respuesta: La altura máxima es 21

Medidas	Paquete 1	Paquete 2	Paquete 3
Lado 1	4	1	3
Lado 2	6	1	4
Lado 3	7	3	2
Peso	10	5	4

Respuesta: La altura máxima es 17.

Desarrolle un programa que empleando programación dinámica calcula la altura máxima de los paquetes indicados. **Recuerde que esta su respuesta tiene validez solo si muestra el arreglo de soluciones.**

Pregunta 3 (10 puntos)

En una embotelladora de bebidas gaseosas, se tienen M líneas de producción, cada una de las cuales presenta estas cuatro máquinas:

1. Llenadora de líquido: que es la máquina que introduce en la botella, el líquido de la gaseosa
2. Gasificadora: que es la máquina que introduce en la botella, el compuesto de carbono para que “tenga gas” la bebida
3. Tapadora: que es la máquina que coloca la chapa o la tapa “rosca” dependiendo del tipo de botella (plástico o vidrio)
4. Etiquetadora: que es la máquina que coloca la etiqueta de la marca de la gaseosa en la botella, dependiendo de lo indicado en la orden de producción.

Por otro lado, se tienen N ordenes de producción de bebidas gaseosas, donde cada una de ellas presenta:

- a. Cantidad a producir
- b. Tipo de embotellado: plástico o vidrio
- c. Marca de la gaseosa.

Cada una de las cuatro máquinas se demoran los siguientes tiempos:

1. Llenadora de líquido: 2 minutos en verter el líquido, por botella.
2. Gasificadora: 1 minuto para gasificar el contenido de la botella, por botella
3. Tapadora: 1.5 minutos para poner la chapa o la tapa “rosca”
4. Etiquetadora: 1.5 minutos en colocar la etiqueta correctamente.



Se pide resolver el problema con un algoritmo voraz indicando, en qué orden se deben procesar las N órdenes para minimizar el tiempo acumulado de trabajo (conocido en la literatura como makespan).

Para resolver el problema considere los siguientes datos:

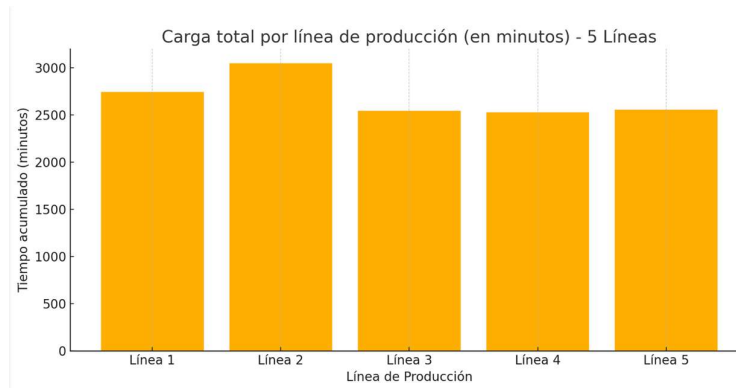
- M = 5 líneas de producción cada una con las 4 máquinas
- N = 20 órdenes de producción con los siguientes datos

ID de Orden	Cantidad de Botellas
1	103
2	58
3	88
4	126
5	195
6	90
7	54
8	195
9	124
10	113
11	193
12	55
13	97
14	97
15	169
16	50
17	167
18	74
19	79
20	109

Línea, Ordenes asignadas, Carga total (minutos)

Línea 1, Orden1(103), Orden8(195), Orden16(50), Orden20(109),2742.0
 Línea 2, Orden2(58), Orden6(90), Orden11(193), Orden17(167), 3048.0
 Línea 3, Orden3(88), Orden7(54), Orden10(113), Orden15(169),2544.0
 Línea 4, Orden4(126), Orden9(124), Orden13(97), Orden18(74),2526.0
 Línea 5, Orden5(195), Orden12(55), Orden14(97), Orden19(79),2556.0

Tiempo acumulado de trabajo = 3048 minutos (makespan)



Al finalizar el examen, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para este examen.

Profesores del curso:

Manuel Tupia
Rony Cueva
Igor Siveroni

San Miguel, 20 de mayo del 2025