

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMOS AVANZADOS

2da. práctica (tipo B)
(Primer Semestre 2025)

Duración: 1h 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, 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_LAB2_P#` (donde # representa el número de la pregunta a resolver)

Pregunta 1 (5 puntos)

Una empresa recibe un grupo de tareas para realizar diariamente en una planta, cada tarea demanda una cantidad de horas. En la actualidad la empresa trabaja con turnos de 8 horas que son consecutivos: turno mañana, tarde y noche. Por tal motivo se desea calcular la cantidad de horas máximas que se puede asignar en cada turno, primero para el turno mañana, luego para el turno mañana y tarde, y finalmente el turno mañana, tarde y noche. Debe considerar que solo se pueden tomar tareas completas, y si una tarea no se puede realizar se envía a otra planta.

Por ejemplo, si se tienen 5 tareas con tiempos tales como: 1, 3, 6, 10, 16

- Si se usa solo el turno mañana las tareas como máximo pueden sumar 7, sin sobrepasar las 8 horas del turno.
- Si se usan dos turnos mañana y tarde las tareas como máximo suman 16, sin sobrepasar las 16 horas de los 2 turnos.
- Si se usan los tres turnos las tareas como máximo suman 23, sin sobrepasar las 24 horas de los 3 turnos.

Desarrolle un programa basado en programación dinámica que calcule el máximo de horas a

partir de un grupo de tareas cuando se considera uno, dos y los tres turnos. Puede emplear los datos que se muestran en los ejemplos.

Pregunta 2 (5 puntos)

Una empresa de exportación transporta diversos productos al extranjero, por lo cual cuenta con un contenedor que puede llevar como máximo 10 artículos. Cada producto tiene un peso determinado (p). Debido a temas de aduana todos los productos elegidos deben ser diferentes y como máximo pueden transportar M productos al extranjero. Además, por restricciones del puerto, el peso total de los productos no debe exceder un peso definido como W . Considere $M \leq 10$.

Ejemplo de ingreso de datos:

$M = 5$

$W = 25$

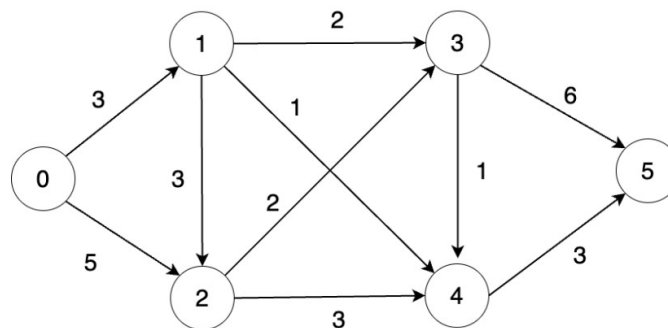
| Artículo | Peso (p) |
|----------|--------------|
| 1 | 7 |
| 2 | 2 |
| 3 | 5 |
| 4 | 5 |
| 5 | 10 |

Resultado: No es posible llegar al peso 25. El peso máximo factible es 24.

Implemente un programa que empleando programación dinámica indique si es o no factible llegar al peso W , debe mostrarse en pantalla un mensaje. En caso no es posible llegar al peso W solicitado debe el obtener el peso máximo factible.

Pregunta 3 (10 puntos)

La empresa para la que trabajas maneja redes de distribución compuestas por un almacén central y una serie de almacenes intermedios conectados por rutas predeterminadas. Las redes de distribución que maneja la empresa estarán representadas por figuras como la siguiente:



Estas redes tienen las siguientes características:

- Cada punto de la red (nodo) ha sido asignado un número.
- El almacén central, o nodo de inicio, está marcado por el número 0.

- Las flechas en el grafo indican si dos nodos de la red esta conectados directamente. Estas conexiones son de un solo sentido. Además, las flechas están anotadas por el tiempo que toma ir de un punto a otro.
 - La red no contiene ciclos.
Además, los nodos de la red han sido numerados de tal manera que toda ruta directa (arista) cumple la siguiente propiedad: Si la arista conecta los nodos j y k , entonces $j < k$. Verificar si esto es cierto en la figura de arriba.
- a) El encargado de logística de la empresa nos pide diseñar un algoritmo que nos permita calcular las rutas más rápidas (que tomen menos tiempo) que conecten al almacén con el resto de los puntos de la red. Se nos pide usar programación dinámica. (5.0 puntos)

Input: Como input se tendrá una matriz indicando los tiempos (y dirección) asociados a cada conexión directa presente en la red. La ausencia de una conexión directa entre 2 puntos será representada por el valor 0. Por ejemplo, la matriz correspondiente a la red de la figura es:

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 3 | 5 | 0 | 0 | 0 |
| 1 | 0 | 0 | 3 | 2 | 1 | 0 |
| 2 | 0 | 0 | 0 | 2 | 3 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 6 |
| 4 | 0 | 0 | 0 | 0 | 0 | 2 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |

Observación: Podemos notar que nuestra matriz es triangular superior y con 0's en las diagonales. Esto refleja el hecho que la numeración de los nodos cumple con la siguiente propiedad: para cada conexión directa (j,k) , $j < k$. Esta propiedad nos será útil para el diseño de nuestros algoritmos usando programación dinámica.

Uso: Se nos pide, dado un punto n de input, imprimir la distancia de la ruta más rápida del almacén 0 (origen) al punto n , y los nodos que componen la ruta.

Así, por ejemplo, para el nodo **5** tenemos:

Distancia mínima de 0 a 5: 7
ruta mínima de 0 a 5: 0 1 4 5

Y para el nodo **3**:

Distancia mínima de 0 a 3: 5
ruta mínima de 0 a 3: 0 1 3

- b) Luego de haber completado la primera tarea, el encargado de logística decide que, en realidad, necesita saber las rutas más rápidas que conecten todos los puntos de la red. Para esto, nos pide implementar un algoritmo usando la técnica de programación dinámica que, dados dos puntos de la red, nos imprima la ruta más rápida que los conecte. (5.0 puntos)

Por ejemplo:

Distancia mínima de 1 a 5: 4
ruta de 1 a 5: 1 4 5

Si no existiese tal ruta, imprimir:

ruta de x a y: inexistente

Para que las soluciones sean válidas en las preguntas debe mostrar el arreglo o matriz de soluciones parciales, recuerde que solo debe emplear iteraciones, si alguna respuesta emplea recursión la pregunta queda anulada.

Al finalizar el laboratorio, 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 laboratorio.

Profesores del curso:

Rony Cueva
Manuel Tupia
Igor Siveroni

San Miguel, 26 de abril del 2025