

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMOS AVANZADOS

4ta. práctica (tipo B)
(Segundo Semestre 2024)

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_LAB4_P#` (donde # representa el número de la pregunta a resolver)

Pregunta 1 (10 puntos)

Se tienen varios objetos de diferentes tamaños que tienen que ser empaquetados en cajas (o contenedores, bins) que tienen un límite de capacidad, siendo que todas las cajas son iguales. La idea es minimizar la cantidad de cajas necesarias, aprovechando al **máximo el espacio disponible en cada una de ellas**.

Ejemplo: Supongamos que tenemos los siguientes objetos con sus respectivos pesos en kilos: 4, 8, 1, 4, 2, 1. Y tenemos cajas con una capacidad máxima de 10 kg. Tenemos que empaquetar dichos objetos con el menor número posible de cajas.

Una solución sería:

- Contenedor 1: [8, 2]
- Contenedor 2: [4, 4, 1]
- Contenedor 3: [1]

Se pide elaborar un algoritmo **GRASP Construcción** que permita recibir una lista de objetos a empaquetar y la capacidad de los contenedores y que devuelva la cantidad de contenedores

utilizados y qué objetos se colocaron en cada uno de ellos. Puede ordenar los paquetes si lo desea.

Pregunta 2 (10 puntos)

La empresa Pedidos YAP necesita contar con un algoritmo GRASP para la optimización de sus entregas, se sabe que los repartidores ganan un monto por cada entrega realizada y muchas veces si van a un punto determinado no pueden ir a otro debido a restricciones de tránsito. Por tal motivo el algoritmo debe maximizar la ganancia para cada repartidor. A continuación, un ejemplo:

Los repartidores siempre salen del punto A

Origen	Destino
A	B C D E H
B	C D F H
C	F
D	-
E	C F G
F	-
G	-
H	-

Se sabe que el pago por realizar una entrega en cada punto es el siguiente:

A	B	C	D	E	F	G	H
-	20	30	40	40	40	10	50

La respuesta para este ejemplo, con un $\alpha=0.4$ y 10,000 iteraciones es:

Ganancia = 110

Puntos seleccionados: E C F

Desarrollar un algoritmo GRASP Construcción que ayude a obtener la máxima de ganancia, puede emplear los datos de ejemplo para sus pruebas.

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:

Manuel Tupia
Rony Cueva

San Miguel, 9 de noviembre del 2024