



INTELIGENCIA ARTIFICIAL
Práctica Calificada Nro. 1

Profesor: Diego Rojas Sihuy

Duración: 180 minutos

Secciones: _____

No escribir en estos recuadros

Pregunta 1	Pregunta 2	Nota
10,0	10,0	20,0

Instrucciones:

- El estudiante no puede hacer uso de apuntes de clase, libros, etc.
- **No puede hacer uso de usb o cualquier medio de almacenamiento digital.**
- Todo apunte es personal, está prohibido prestar o intercambiar apuntes.
- Solo puede usar **Google Colab y/o windows explorer** durante el examen
- El **proyecto deberá ser almacenado en la carpeta y con el nombre que indique el profesor.**
- Anote en el recuadro el número de la PC

Número de PC:

Este dato es de vital importancia, si Ud. no lo indica, no habrá manera de saber cuál es su examen y por tanto no podrá ser calificado y Ud. tendrá como nota CERO.

- Durante el examen, sólo puede utilizar un Entorno de Python (Google Colab) y el Explorador del Windows.

Al finalizar del examen el estudiante deberá entregar el texto del examen

Caso1: Optimización de Picking Térmico y Energético en Almacenes

La empresa FrostLogistics gestiona un almacén especializado en alimentos congelados que requiere mantener productos sensibles dentro de rangos de temperatura estrictos.

Cada celda del almacén tiene:

- Una temperatura (°C)
- Un costo de energía asociado a mantener dicha temperatura (cuanto más frío, más energía)

La empresa desea trazar una ruta de picking eficiente desde la zona de ingreso (0,0) hasta la zona de despacho (última celda abajo a la derecha).

El objetivo principal es minimizar cambios drásticos de temperatura, pero también considerar la energía utilizada.

- Solo se permiten movimientos hacia abajo o hacia derecha (estructuras físicas del almacén).
- Se busca optimizar el recorrido bajo distintos criterios.

Se proporciona la siguiente matriz:

Matriz de Temperaturas (°C)		
-18	-19	-20
-16	-17	-20
-15	-15	-20

Matriz de Costos Energéticos		
5	6	7
4	5	7
3	3	7

Pregunta 1: Heurística Greedy Basada en Temperatura (5 puntos)

Desarrollar un código en Python explicando paso a paso según el método de heurística para moverse del inicio al destino minimizando el cambio absoluto de temperatura respecto al objetivo final (-20°C).

En cada movimiento, podrá elegir solo entre moverse hacia abajo o derecha. La decisión en cada paso debe ser basada únicamente en minimizar $|\text{temperatura actual} - \text{temperatura final}|$.

Dibuje el camino final o descríballo paso a paso.

Pregunta 2: Aplicación del Algoritmo A* considerando Energía (5 puntos)

Desarrollar un código en Python explicando paso a paso aplicando un esquema del algoritmo A* para encontrar la ruta desde el inicio hasta el final que minimice la combinación del costo energético total más la desviación de temperatura final.

Defina claramente:

- Cómo calcula $g(x)$
- Cómo calcula $h(x)$

Resuelva teniendo las siguientes consideraciones:

- $g(x)$: suma de costos energéticos acumulados + diferencia de temperatura en el camino recorrido.
- $h(x)$: distancia Manhattan hasta el final considerando penalizaciones de temperatura y energía estimadas.

Caso 2: Optimización del Equipamiento de Entregas de Productos Sensibles

La empresa FrostDeliveries realiza entregas rápidas de productos congelados a supermercados.

Cada repartidor tiene una mochila térmica especial que:

- Tiene una capacidad máxima de peso de 10 kg.
- Solo puede cargar un número limitado de productos que deben ser escogidos cuidadosamente.

La empresa dispone de 8 productos disponibles para el repartidor:

Producto	Peso (kg)	Valor de conservación (importancia)
Helado premium	2	9
Carne congelada	3.5	8
Vegetales congelados	2.5	6
Pescado	4	7
Yogurt helado	1	5
Queso crema	0.5	4
Mantequilla	0.8	3
Paletas de hielo	1.5	5

El objetivo no es solo llenar la mochila, sino maximizar el "valor de conservación total" respetando la restricción de peso.

No todos los productos deben ser necesariamente elegidos.

Pregunta 3: Optimización usando Hill Climbing (5 puntos)

Desarrolle un programa en Python que utilice el algoritmo Hill Climbing para encontrar una combinación de productos que:

- No exceda los 10 kg de peso total.
- Tenga el mayor valor de conservación posible.

El algoritmo debe:

- Iniciar con una combinación aleatoria válida de productos (no sobrepasar los 10 kg).

- Generar vecinos agregando o quitando un producto.
- Moverse únicamente hacia combinaciones vecinas que mejoren el valor total de conservación.
- Terminar cuando ninguna mejora sea posible.

Requisitos:

- Mostrar la combinación final de productos seleccionados.
- Indicar el peso total, el valor total, y el número de movimientos realizados.

Pregunta 4: Optimización usando Simulated Annealing (5 puntos)

Desarrolle un programa en Python que resuelva el mismo problema utilizando Simulated Annealing.

El algoritmo debe:

- Iniciar con una combinación aleatoria válida.
- Permitir movimientos hacia combinaciones peores con una probabilidad que depende de una temperatura de control.
- Reducir la temperatura progresivamente en cada iteración.
- Aceptar o rechazar nuevos estados según la mejora o la probabilidad de escape de óptimos locales.

Requisitos:

- Mostrar el proceso de evolución del valor total a medida que la temperatura baja.
- Imprimir la combinación final, peso total, valor total, y el número de iteraciones realizadas.