

<b>Estructura de Datos y Algoritmos</b> <b>Grado Ingeniería Informática</b> <b>Doble Grado Ingeniería Informática y</b> <b>Administración de Empresas</b> <b>Universidad Carlos III de Madrid</b> <b>CURSO 2020-2021</b>	 
---	---



**Convocatoria Ordinaria. 24 Mayo 2021.**

**Nombre:**

Instrucciones:

- Escribe tu nombre en esta hoja y en las hojas de cuadro.
- La duración del examen son 120 minutos.
- Los móviles deben estar apagados y nunca pueden estar encima de la mesa.
- Utiliza las hojas de cuadros para preparar tu solución. Cuando estés conforme con tu solución, pásala a limpio. ÚNICAMENTE PODRÁS ENTREGAR DOS HOJAS DE CUADROS.
- Cuida al máximo la presentación de tu solución. Es imprescindible que tu código sea claro, bien ordenado y limpio.
- No puedes pedir tipex, bolígrafo, etc a tus compañeros. Si necesitas algo, pídeselo al profesor.
- Debido al Covid, si tienes alguna pregunta, levanta tu mano y espera a ser atendido. Tendrás que preguntar en voz alta.
- No puedes salir del examen antes de que pasen 30 minutos.
- No está permitido ir al baño.
- Cuando termines, entrega el cuadernillo y las dos hojas de cuadros al profesor.

**Problema 1:** Implementa el algoritmo **mergesort** para ordenar una lista doblemente enlazada. Puedes añadir funciones auxiliares que te ayuden a resolver tu problema. Las funciones de DList no deben implementarse, es decir, puedes invocarlas directamente. No está permitido usar estructuras de Python (listas de python, diccionarios, etc) para resolver el problema.

```
from dlist import DList

def mergesort(l):
    result=DList()
    ...
    return result
```

**Problema 2:** La clase MyGraph es una implementación basada en diccionarios para representar un grafo dirigido no ponderado. Para simplificar el problema, supondremos que los vértices son números enteros no negativos (0,1,2,...). Implementa la función *minimumPath*, que reciba dos vértices, *start* y *end*, y devuelve una lista de Python con los vértices que forman el camino mínimo desde start a end, ambos inclusive. La función minimumPath debe utilizar el algoritmo de camino mínimo de Dijkstra. Como el grafo es un grafo no ponderado, se considera como camino mínimo aquel que tenga un menor número de aristas. Está permitido utilizar estructuras de Python como las listas o los diccionarios.

```
class MyGraph:
    def __init__(self,n):
        """Crea un grafo con n vértices (0,1,...,n-1)"""
        self.vertices={}
        for i in range(n):
            self.vertices[i]=[]

    def addConnection(self,i,j):
        ...

    def minimumPath(self, start, end):
        ...
```