

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

uc3m | **Universidad**
Carlos III
de Madrid



Convocatoria Ordinaria. 24 Mayo 2021.

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):
        ...
```