

# Matemàtica Discreta - Seminari 4 - Camins òptims

## Funcionament dels seminaris

El primer i primordial que heu de saber és:

---

**ÉS MOLT IMPORTANT LLEGIR BÉ ELS ENUNCIATS I CONTESTAR EL QUE ES DEMANA DE LA MANERA QUE ES DEMANA, NO UNA ALTRA COSA.**

---

Aquest és el motiu més habitual de problemes tant als seminaris com als exàmens.

Durant aquests seminaris intentarem aplicar, d'una forma pràctica, aquells coneixements que anem adquirint a teoria i problemes de l'assignatura. Per seguir els seminaris necessiteu un ordinador, un editor de textos i algun tipus de IDE que us permeti programar i executar Python3.

Cada seminari es compondrà d'una sèrie de preguntes i d'algun exercici de programació. Per facilitar la feina, per a cada seminari tindreu disponible un esquelet de programa que heu d'omplir i d'un conjunt de tests per validar que el vostre programa funciona. Per tant, sempre disposareu d'una carpeta anomenada esquelet amb almenys tres fitxers. Un main per fer proves o crear objectes concrets, un o més fitxers que heu d'omplir i on tindreu pistes sobre el que heu de fer i un fitxer de tests.

**Per a començar sempre heu de duplicar la carpeta esquelet i dir-li 'solucio' i en aquesta carpeta és on heu de fer els canvis. La carpeta esquelet no l'heu de tocar per res.**

El fet que el programa passi el conjunt de tests donat, no vol dir que sigui correcte. Primer, perquè els tests donats no comproven tots els casos és a dir, es valorà la capacitat de tractar casos extrems o no contemplats, i a més, es valorarà que el codi segueixi els conceptes d'Eficiència, Escalabilitat i Llegibilitat. En canvi, no passar el conjunt de tests sí que es considerarà un seminari no correcte.

Els lliuraments es compondran d'un pdf amb les respostes a les preguntes teòriques, d'un arxiu README.md amb les instruccions per a l'execució i els arxius .py necessaris per a la seva execució. Tot comprimit en un zip o un tar.gz.

## Exercici 1 - Afegir costos a les arestes

Primerament, hem de permetre que el nostre graf representi costos associats a cada aresta. Per a fer-ho, necessitem modificar algunes funcions de la classe graph.py. Si us hi fixeu, cada node té una llista d'arestes anomenada edges. Aquesta llista conté altres nodes. Hem de modificar l'estructura perquè aquesta llista contingui tuples (conjunts de dos elements) on el primer sigui el cost de l'aresta, i el segon el node.

Seguiu l'arxiu graph.py buscant les funcions que estan marcades per a modificar i feu els canvis necessaris. En python, una tupla es pot representar simplement com (cost, node).

## Exercici 2 - Escapar d'Invernalnia

Imagineu que sou els darrers hereus a Reis/Reines del Nord, amb capital a Invernalnia. Malauradament, a causa de conxorxes i traïcions, teniu els exèrcits dels Lannister a les portes de la ciutat. Com que és improbable que la ciutat aguantí, cal traçar un pla de fugida. Reunits amb els vostres darrers aliats, voleu disposar de tota la informació necessària per prendre la millor de les decisions. Se us acudeix calcular el mínim de dies que us calen per arribar a totes les capitals i, amb aquesta informació, prendre la millor decisió.

A l'arxiu main.py, trobareu una funció anomenada crea\_joc\_de\_trons(). Aquesta funció crea el mapa de les principals capitals de Ponent de la coneguda saga de ciència-ficció. Cada capital és un node del nostre graf i cada aresta té un pes que representa els dies que es tarda per arribar d'una capital a l'altra. Evidentment, hi ha rutes a peu i rutes per mar i, per tant, poden haver-hi combinacions que tot i recórrer més kilòmetres, siguin més ràpides.

Què heu de fer:

- A mà: feu la solució del Dijkstra representat per aquest graf.
- Implementeu la funció Dijkstra al arxiu cerques.py.

Pistes:

- `sys.maxint` us dona el valor màxim que pot tenir un enter en python.
- És recomanable programar-se una funció auxiliar que donades tres llistes (nodes, costos i pivots) us selecciona el següent pivot i us en retorna també el cost. En teniu la signatura al fitxer.

### Exercici 3: Recuperar el recorregut

Amb la solució donada al exercici 2, només sabem el cost del camí mínim, però no podem recuperar el recorregut que hem de seguir per obtenir aquest camí mínim.

Què heu de fer:

- A mà: com recuperarieu el camí mínim?
- Modifiqueu l'algoritme implementat perquè, a més de retornar la llista de costos ordenada seguint l'ordre dels nodes del graf, també recuperi el node del qual es prové. Per exemple, si només tinguéssim els nodes 'Invernalía', 'Aguasdulces' i 'Nido de Águilas', el resultat hauria de ser:
  - Nodes del graf: ['Invernalía', 'Aguasdulces', 'Nido de Águilas']
  - Costos: [0, 6, 8]
  - Recorreguts: ['Invernalía', 'Invernalía', 'Aguasdulces']

### Exercici 4 (opcional): Floyd

Ara imagineu que volem tenir en compte també els moviments de la resta d'exèrcits. Hauríem de tenir una matriu de costos d'anar de qualsevol ciutat a qualsevol altra.

Què heu de fer:

- A mà: calculeu floyd del graf donat
- Implementeu la funció floyd que donat un graf, retorna la matriu (llista de llistes) amb els costos d'anar de qualsevol ciutat a qualsevol altra.

---

**Lliurament:** Heu d'entregar, per grups, un tar.gz o zip amb les preguntes respostes i el codi al CV abans del 28 de maig a les 23:59.