

# PROGRAMACIÓ AVANÇADA

## TRIMESTRE 1. CURS 2022/23

### PRÀCTICA 5

.....

**Objectiu 1: Resolució d'un exercici aplicant la tècnica del Backtracking.**

**Objectiu 2: Resolució d'un exercici aplicant la tècnica de Voraç.**

**Durada:** Una sessió

**Lliurament:** Llistat imprès dels fonts i penjar el projecte al Moodle

**Data Lliurament:** Al final del trimestre: **20 de Desembre a les 23:50h**

En aquesta pràctica es resoldrà un mateix exercici aplicant les dues tècniques. **Dues solucions alternatives pel mateix enunciat.**

#### Enunciat

Es tracta d'implementar un programa per trobar el **nombre mínim de compartiments d'emmagatzematge** que ha de tenir una empresa del sector químic per garantir la seguretat, evitant l'explosió si algun producte que emmagatzema entra en contacte en d'altres. Els productes químics poden reaccionar entre sí donant lloc a explosions si entren en contacte. Com a precaució l'empresa vol dividir el magatzem en compartiments i emmagatzemar productes incompatibles en compartiments diferents.

La solució del problema ha de tenir una **classe Producte** amb atributs per emmagatzemar les dades de cada producte, la identificació d'aquest i la llista de productes amb qui reacciona i amb conseqüència s'hauran d'emmagatzemar separatament. Determineu la millor manera d'emmagatzemar aquestes incompatibilitats, trieu el magatzem més adient d'entre les col·leccions donades a l'API de Java. Afegiu els mètodes necessaris tenint en compte que els atributs han de ser privats i del magatzem no es podrà realitzar un getter, s'hauran d'obtenir els productes incompatibles d'un en un.

#### **Solució 1. Aplicant la tècnica del Backtracking**

**Aplicant la tècnica del backtracking** cal trobar **el número mínim de compartiments que ha de tenir el magatzem de l'empresa** per emmagatzemar N productes químics diferents i en quin s'emmagatzemarà cadascun d'ells.

La solució s'ubicarà dins una segona classe anomenada **SolucioBack** que contindrà el mètode que fa backtracking, amb els mètodes privats ajudants i el procediment **main** per llançar l'execució. Aquesta classe tindrà atributs:

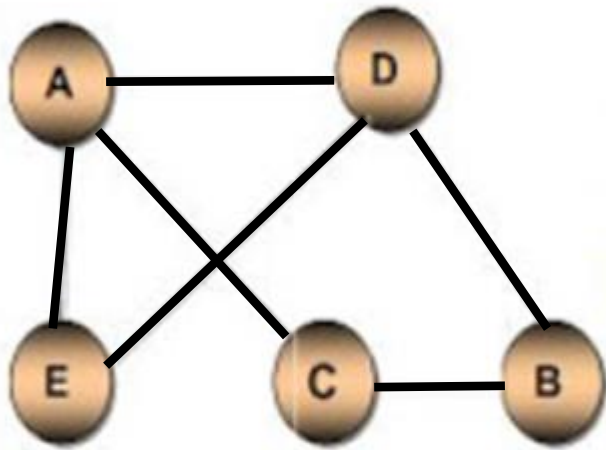
- ✓ Un magatzem amb tots els objectes Producte que ha d'emmagatzemar l'empresa.

- ✓ Els magatzems necessaris per emmagatzemar les solucions del backtracking.
- ✓ Altres atributs per controlar si la millor solució trobada fins aleshores és o no millor a la que s'està construint.
- ✓ Un atribut per emmagatzemar la **matriu d'adjacència** que s'ha de construir a partir de les incompatibilitats dels productes. Atribut que consultarà el backtracking per determinar si una decisió és o no acceptable, evitant així de fer cerques costoses als magatzems de cada producte.

### **Exemple. Matriu d'adjacència**

Suposeu que els productes que s'han d'emmagatzemar són: A, B, C, D, E

**Les relacions d'incompatibilitat** entre productes les marco amb una línia en el següent gràfic:



El gràfic indica que els producte A reacciona amb el D, C i E en conseqüència no es poden emmagatzemar en el mateix compartiment. El D reacciona amb el A, E i el B. El E amb el A i D. El C amb el A i B. I el B amb el C i D.

Aquest gràfic s'anomena graf. Les rodones, anomenats vèrtexs, representen el productes i les línies entre elles, que s'anomenen arestes, la relació d'incompatibilitat. L'emmagatzematge d'aquest graf usant matriu d'adjacència és la següent:

	0	1	2	3	4
0	F	F	T	T	T
1	F	F	T	T	F
2	T	T	F	F	F
3	T	T	F	F	T
4	T	F	F	T	F

El vector i la matriu estan relacionades per índex. Les dades del producte A que està emmagatzemat a la posició d'índex zero al vector estan ubicades a la fila i columna

zero de la matriu. Un valor de cert indica que són incompatibles i un de fals que no ho són.

Adoneu-vos de què la matriu és sempre simètrica. El vostre programa un cop ha obtingut les dades del problema a resoldre, creant els objectes Producte amb el seu magatzem d'incompatibilitats **ha de crear la matriu d'adjacència que serà consultada al llarg del backtracking.**

El problema s'ha de resoldre aplicant la tècnica del backtracking, i com és un problema de **minimització** cal aplicar la variant de trobar la millor solució. **Obligatòriament s'ha d'aplicar l'esquema treballat a classe de teoria.**

**Prèviament a la codificació, obligatòriament s'ha de fer l'anàlisi del problema.**

**Anàlisi del problema. Qüestions a contestar prèviament a la implementació**

- 1.- Per què l'esquema de backtracking és aplicable per a resoldre aquest enunciat.
- 2.- Indica quina pregunta et fas en cada nivell de l'arbre. Quines són les possibles respostes d'aquesta pregunta (domini).
- 3.- Quin és el criteri per determinar si una decisió és o no acceptable.
- 4.- Quin és el criteri per determinar si un conjunt de decisions és o no completable.
- 5.- Quin és el criteri per determinar si un conjunt de decisions són o no solució.
- 6.- **Dibuixeu** l'espai de cerca del problema, és a dir l'arbre que ha de recórrer la tècnica del backtracking, indica quina és l'alçada i l'amplada i si són valors exactes o valors màxims. Usaràs marcatge?.
- 7.- Quin és el criteri per determinar si una solució és o no millor a una altra ja trobada prèviament.

**Qualsevol implementació que NO s'ajusti a l'espai de cerca que indiqueu en la contestació de les preguntes prèvies es considerarà INCORRECTE, malgrat funcioni.**

**Respecte l'entrada de dades**, cal que sigui **aleatòria**. Únicament a l'usuari se li demanarà el nombre de productes que ha d'emmagatzemar l'empresa. La resta de dades, identificació de cada producte, i **reaccions entre ells han de ser aleatòries**, però "coherent", la relació ha de ser simètrica.

L'execució del **main** ha de ser similar a:

Indica quants productes diferents ha d'emmagatzemar l'empresa?

5

Valor entrat  
per l'usuari

Ara es procedeix a generar les dades aleatòries:

Producte 1 (P1) amb incompatibilitats amb els productes: 3, 4, 5

Producte 2 (P2) amb incompatibilitats amb els productes: 3,4

Producte 3 (P3) amb incompatibilitats amb els productes: 1, 2

Producte 4 (P4) amb incompatibilitats amb els productes: 1, 5, 2

Producte 5 (P5) amb incompatibilitats amb els productes: 1, 4

Valors generats  
aleatòriament que  
han de ser coherents  
(simetria)

Matriu d'adjacència:

	P1	P2	P3	P4	P5
P1	F	F	T	T	T
P2	F	F	T	T	F
P3	T	T	F	F	F
P4	T	T	F	F	T
P5	T	F	F	T	F

Matriu d'adjacència generada  
pel programa

No és necessari que surtin les  
divisions vertical ni horitzontals  
en la visualització a pantalla

Solució trobada necessita 3 compartiments.

L'assignació és la següent:

Producte 1 al compartiment 1  
Producte 2 al compartiment 1  
Producte 3 al compartiment 2  
Producte 4 al compartiment 3  
Producte 5 al compartiment 3

Distribució trobada pel backtracking. Si  
hi ha més d'una solució és irrellevant  
la trobada

Un cop demanades les dades i generades aleatòriament les incompatibilitats s'ha de mostrar la matriu d'adjacència del problema i la millor solució del problema. Dades a visualitzar:

- ➔ Número de productes a emmagatzemar. *Especificat per l'usuari.*
- ➔ Incompatibilitats de cada producte. *Generades aleatòriament. Valor mínim zero i màxim numero de productes -1, és a dir cap incompatibilitat i incompatibilitats amb tots els altres productes (no pot amb ell mateix).*
- ➔ Matriu d'adjacència (relació d'incompatibilitat). *Generada pel programa.*
- ➔ Número mínim compartiments. *Calculat pel backtracking.*
- ➔ Compartiment assignat a cada producte. *Calculat pel backtracking.*

## Consideracions

1.- Es valorarà l'anàlisi descendent, es desitjable que cada part de l'esquema sigui implementat mitjançant un procediment.

- 2.- Adoneu-vos de què aquest problema sempre té solució, serà la pitjor i vindrà donada per la que assigna un compartiment diferent a cada producte.
- 3.- La matriu d'adjacència serà sempre simètrica, la relació d'incompatibilitat és simetria, és a dir, si A reacciona amb B també B reacciona amb A. La diagonal principal sempre emmagatzemarà un valor de fals.
4. El mètode que implementa el backtracking ha de tenir **el menor número de paràmetres possibles**.

## **Solució 2. Aplicant la tècnica Voraç**

**Aplicant la tècnica greedy** cal trobar **el número mínim de compartiments que ha de tenir el magatzem de l'empresa** per emmagatzemar N productes químics diferents i en quin s'emmagatzemarà cadascun d'ells.

La solució s'ubicarà dins una tercera classe anomenada **SolucioVoraç** que contindrà el mètode **voraç**, amb els mètodes privats ajudants i el procediment **main** per llançar l'execució. Aquesta classe tindrà atributs:

- ✓ Un magatzem amb tots els objectes **Producte** que ha d'emmagatzemar l'empresa.
- ✓ Els magatzems necessaris per emmagatzemar la solució **voraç**.
- ✓ Un atribut per emmagatzemar la **matriu d'adjacència** que s'ha de construir a partir de les incompatibilitats dels productes. Atribut que consultarà el mètode **voraç** per determinar si una decisió és o no acceptable.
- ✓ Podeu afegir atributs i mètodes, els que considereu adients.

**Prèviament a la codificació, obligatòriament s'ha de fer l'anàlisi del problema.**

**Anàlisi del problema. Qüestions a contestar prèviament a la implementació**

- 1.- Perquè aquest problema pot resoldre's aplicant la **tècnica voraç**?
- 2.- Quins són els candidats del problema?
- 3.- Quina funció de selecció aplicarà el vostre algorisme? **Cal trobar la millor**
- 4.- La vostra funció de selecció, garanteix trobar sempre la millor solució? Per què?

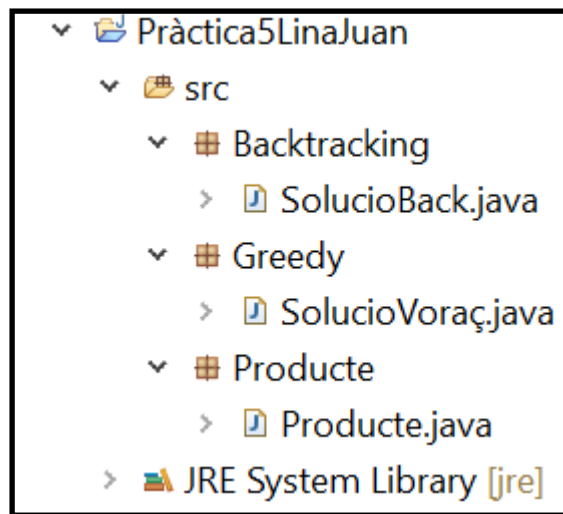
**Respecte l'entrada de dades**, idèntica a la versió prèvia, ha de ser **aleatòria**. Únicament a l'usuari se li demanarà el nombre de productes que ha d'emmagatzemar l'empresa. La resta de dades, identificació de cada producte, i **reaccions entre ells han de ser aleatòries**, però "coherents", la relació ha de ser simètrica.

L'execució del **main** ha de visualitzar la mateixa sortida que en la resolució amb backtracking, ara amb la solució trobada per la tècnica **voraç**.

## **Organització**

Creeu un nou projecte per aquesta pràctica. **Els noms dels vostres projectes, en les pràctiques d'aquesta assignatura han de seguir el següent patró: PràcticaXCognom1Nom1&&Cognom2Nom2**. En aquesta pràctica X és 5. Aquest

projecte ha de tenir els següents paquets. I cada paquet les classes indicades en la següent imatge:



### Què se us subministra?

Fitxer amb l'enunciat de la pràctica.

### Què s'ha de lliurar i com?

S'ha de lliurar la carpeta que conté el **projecte** Eclipse amb el vostre desenvolupament de la pràctica. La carpeta s'ha de lliurar amb tot el seu contingut i comprimida amb ZIP o RAR.

També s'ha de lliurar el **llistat en paper** del codi desenvolupat. El format de lliurament d'aquest codi ha de seguir el patró indicat en la presentació de l'assignatura: amb portada, índex, número de pàgina, tabulació ... En **aquest llistat** cal que indiqueu quina ha estat **la distribució de la feina entre els dos estudiants. És a dir, el grau de participació de cada membre del grup en la realització d'aquesta activitat. Explica les principals dificultats en el seu desenvolupament, valoració del codi lliurat i funcionament de la pràctica.**

També s'ha de lliurar un document imprès amb **l'anàlisi del problema per a cadascuna de les tècniques, contestant les preguntes plantejades en cada cas.** Imprescindible que la implementació s'ajusti a l'anàlisi del problema. Si no és així, malgrat funcioni no es considerarà correcte.

### On s'ha de lliurar?

El lliurament del projecte es farà a través de la plataforma Moodle i no s'acceptarà cap altra via. Feu atenció a la data i hora límit.

**El lliurament en paper es farà directament a la professora a l'inici de l'examen 2, el 22 de Desembre.**

### Quan s'ha de lliurar?

El lliurament es podrà fer fins el **dia i hora indicats**. Tingueu present que a partir d'aquesta hora el sistema bloquejarà, de manera automàtica, la possibilitat de lliurament.

Tots els grups Moodle → 20 de Desembre a les 23:59h

Tots els grups Paper → 22 de Desembre inici examen

### **Pautes de correcció**

#### **Backtracking (8 punts)**

→ Anàlisi previ: 1 punt

→ Backtracking implementació: 7.5 punts

Es valora: anàlisi descendent, ús de mètodes, atributs, classes, eficiència, robustesa, ....

#### **Voraç (2 punts)**

→ Anàlisi previ: 0.5 punts

→ Implementació voraç: 1.5 punts

Es valora: anàlisi descendent, ús de mètodes, atributs, classes, eficiència, robustesa, ....

**Si la implementació no s'ajusta a la anàlisi que heu fet, malgrat funcioni es considerarà incorrecte, la pràctica tindrà una puntuació de zero. També si els esquemes aplicats no són els treballats a classe de teoria.**