

Pràctica 2. *Backtracking.*

Tamany del grup: *màxim grups de 3 alumnes.*

Tipus de correcció: *presencial*

Data entrega: *setmana de la correcció de l'examen final de gener.*

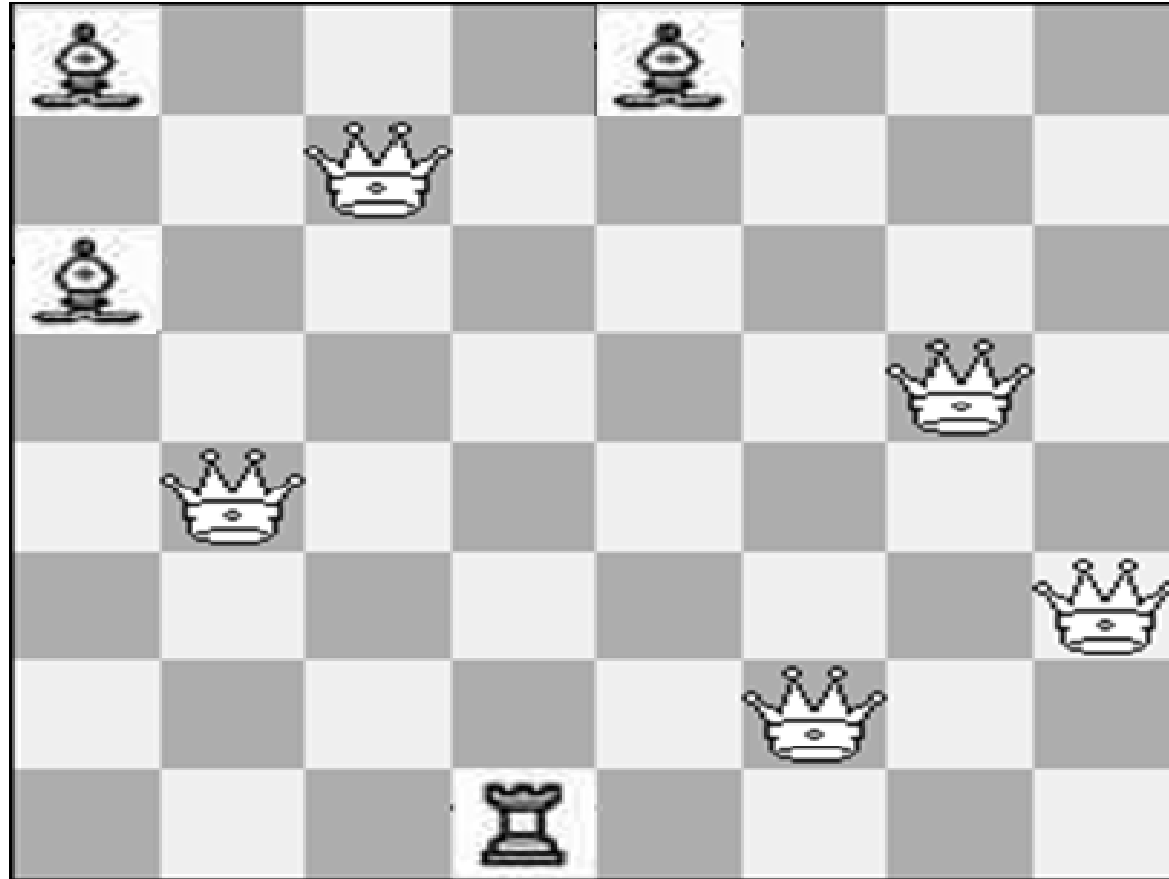
Puntuació: Màxim: 20 puns, Minim: 10 punts; segons l'elecció dels estudiants

Consideracions generals

- A continuació es presenten varies propostes de projectes per a la pràctica 1 de backtracking.
- Cada proposta té una puntuació màxima assignada. El **màxim de puntuació** al que es pot optar són **20 punts**. El **minim** que s'exigeix és de **10 punts**. Això vol dir que un mateix grup d'estudiants pot combinar diferents projectes a realitzar per tal d'assolir la màxima puntuació.
- Els resultats de tots els projectes s'han de mostrar mitjançant una interfície gràfica.
- Les pràctiques es corregiran la setmana de l'examen final de gener, de forma presencial.

Projecte 1. Les peces que no es maten (20 punts)

- Es tracta de desenvolupar un programa que sigui capaç de solventar el següent problema:
- «Col·locar **m** peces de possiblement diferent tipus, a un tauler de dimensió **$n \times n$** , de forma que cap d'elles mati a cap altre de les peces situades al tauler».
- Així, **l'usuari seleccionarà inicialment el tipus de les peces que s'han de col·locar i la quantitat de cada tipus, a més del tamany del tauler**. Per exemple, si l'usuari vol col·locar cinc reines, tres alfils i una torra, veim que $m=9$.
- A la figura següent podeu veure un exemple de solució per a un cas a on tenim cinc reines, tres alfils i una torre on cap de les peces del tauler de 8×8 mata a cap altre.



- L'aplicació a desenvolupar ha de complir els següents requisits obligatoris:
 1. Ha de disposar de les peces següents: **reina, rei, cavall, alfil i torra**, a més d'**una peça inventada** per l'alumne (o grup de d'alumnes).
 2. La solució que donarà l'aplicació final enfront d'un determinat problema ha de ser calculada per un algorisme de ***backtracking***.
 3. El programa ha de disposar d'una presentació dels resultats mitjançant una interfície gràfica que sigui agradable i fàcil d'interpretar, a on es pugui veure la solució final obtinguda per l'aplicació, si existeix. Si no hi ha solució enfront d'un determinat problema s'haurà d'informar a l'usuari d'aquesta conclusió.

- L'activitat es pot fer en grups de 3 persones i ha de contenir una **memòria en paper amb l'esquema (algorisme) recursiu en pseudocodi que s'ha fet servir.**

- **Projecte 2. Trobar 3 solucions pel problema dels moviments del cavall (10 punts).**
- Implementau un programa **backtracking** que trobi 3 solucions al problema dels moviments del cavall en un tauler de $n \times n$, on n és arbitrari i ve definit per l'usuari.
- **Al començament de l'execució del programa, l'usuari definirà el tamany del tauler i la posició inicial del cavall.**
- Les 3 primeres solucions que es trobin s'han de mostrar de manera clara i agradable a l'usuari.
- Si no es possible trobar 3 solucions, llavors el programa ho ha de indicar, mostrant les solucions que ha trobat.
- L'activitat ha de contenir una memòria en paper amb l'esquema (algorisme) backtracking pseudocodi que s'ha fet servir.

- **Pojecte 3. Trobar una solució per el problema de les n reines, on la primera reina és situada per l'usuari en una casella arbitrària del tauler (10 punts).**
- Implementau un programa backtracking que situi n reines en un tauler de $n \times n$ caselles de manera que cap de les reines es pugui matar i on **la posició de la primera reina ve donada per l'usuari**. El moviments de les reines són els mateixos que al joc d'escacs.
- El tamany del tauler, n , vendrà donat per l'usuari al començament de l'execució, així com la **posició a on es situarà la primera reina**.
- Si no és possible trobar una solució, llavors el programa ha de indicar-ho.
- Els resultats s'han de mostrar de manera clara i agradable per l'usuari.
- L'activitat ha de contenir una memòria en paper amb l'esquema (algorisme) backtracking pseudocodi que s'ha fet servir.

- **Projecte 4. Trobar totes les solucions per el problema de les n reines (10 punts).**
- Implementau un programa backtracking que trobi totes les solucions al problema de situar n reines en un tauler de $n \times n$ caselles, de manera que cap de les reines es pugui matar. El moviments de les reines són els mateixos que al joc d'escacs.
- El tamany del tauler, n , vendrà donat per l'usuari al començament de l'execució.
- Si no és possible trobar cap solució, llavors el programa ha de indicar-ho.
- Els resultats s'han de mostrar de manera clara i agradable per l'usuari.
- L'activitat ha de contenir una memòria en paper amb l'esquema (algorisme) backtracking pseudocodi que s'ha fet servir.

- **Projecte 5. Trobar totes les solucions òptimes pel problema de la motxilla 0/1 (10 punts).**
- Implementau un programa backtracking per trobar totes les solucions òptimes al problema de la motxilla 0/1.
- EL nombre, **n**, d'objectes, els seus **pesos** i els **valors** dels mateixos seran definits per l'usuari al començament de l'execució del programa.
- Si no és possible trobar cap solució, llavors el programa ha de indicar-ho.
- Si existeix més d'una solució òptima, llavors el programa les ha de mostrar totes.
- Els resultats s'han de mostrar de manera clara i agradable per l'usuari.
- L'activitat ha de contenir una memòria en paper amb l'esquema (algorisme) backtracking pseudocodi que s'ha fet servir.

- **Projecte 6. Trobar totes les assignació òptimes d'un conjunt de tasques a una plantilla de treballadors (10 punts).**
- Implementau un programa backtracking per trobar **totes les assignacions òptimes** pel problema de l'assignació d'un conjunt de n tasques a un conjunt de n treballadors, de manera que cada treballador tengui una única tasca assignada.
- El nombre, n , de tasques i treballadors i el rendiment $\mathbf{B[i,j]}$ de cada treballador i respecte de la tasca j , els definirà l'usuari al començament de l'execució del programa.
- Si existeix més d'una solució òptima, llavors el programa les ha de mostrar totes.
- Els resultats s'han de mostrar de manera clara i agradable per l'usuari.
- L'activitat ha de contenir una memòria en paper amb l'esquema (algorisme) backtracking pseudocodi que s'ha fet servir.