# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

## Enunciat

Un llibreter acaba d'obrir un negoci de venda de llibres de diferents temàtiques, que s'ofereixen al públic ubicades en diferents prestatgeries.
Consideracions:

- Es vol optimitzar l'espai del local i per tant es pretén minimitzar el nombre de prestatgeries a usar.
- Es imprescindible que tots els llibres d'una mateixa temàtica estiguin ubicats en una mateixa prestatgeria.
- Una prestatgeria pot estar ocupada per llibres de diferents temàtiques mentre tingui espai suficient per tots els seus corresponents llibres.
- Totes les prestatgeries tenen la mateixa capacitat.
- Tots els llibres ocupen el mateix espai.
- Si es dona el cas de què una prestatgeria no té suficient capacitat per contenir tots els llibres d'una única temàtica aquesta no s'ubicarà a la botiga.

El programa en haurà d'informar, indicant les temàtiques que no s'ha pogut ubicar i per tant no es posaran a la venda.

Es vol escriure un **programa voraç** que determini la **distribució de les temàtiques** en les diferents prestatgeries, **minimitzant el nombre** d'elles.

La funció de selecció que heu d'usar és la següent:

- L'estratègia voraç ha de recorre totes les temàtiques a col·locar i ho farà de major a menor número d'unitats de llibres alhora d'assignar la prestatgeria.
- La prestatgeria seleccionada serà aquella que tingui espai lliure suficient per la temàtica completa i que a més deixi el menor espai lliure.
- Si no hi té cabuda en cap de les prestatgeries que ja tenen temàtiques assignades aleshores agafarà una prestatgeria nova (una buida, que no conté res).
- Si la temàtica necessita més espai que la capacitat màxima d'una prestatgeria aquesta temàtica no apareixerà a la solució final.

# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

## Preguntes

1. (0.5 punts) Raona si la funció de selecció indicada trobarà o no la millor solució al problema. Justifica la resposta.
2. (3.5 punts) Afegeix i implementa els mètodes pendents de la classe Aplicacio:
   2.1. Indica els atributs que afegeixes. Important explicar l'ús que en faràs.
   2.2. Inicialitza i/o crea en el constructor els atributs afegits.
   2.3. Completa el main creant l'objecte i invocat al mètode voraç i mostrant a pantalla la solució trobada.
   2.4. Escriu el mètode voraç. Completa el prototipus indicant retorn i paràmetres. Imprescindible que invoqui al mètode que implementa la funció de selecció.
   public ??? solucio(????????)
   2.5. Escriu el mètode funció de selecció. Completa el prototipus indicant retorn i paràmetres. Imprescindible que implementi l'estratègia indicada en l'enunciat com a funció de selecció.
   private ??? funcioSeleccio(????)
   2.6. Redefineix el mètode toString. La visualització ha de ser similar al mostrat als exemples.
3. (1 punt) Aquest problema també pot esser resolt aplicant la tècnica del backtracking.
   Fes l'anàlisi d'aquest problema contestant totes les preguntes indicades en l'exercici anterior, exercici 1 part a i b.

# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

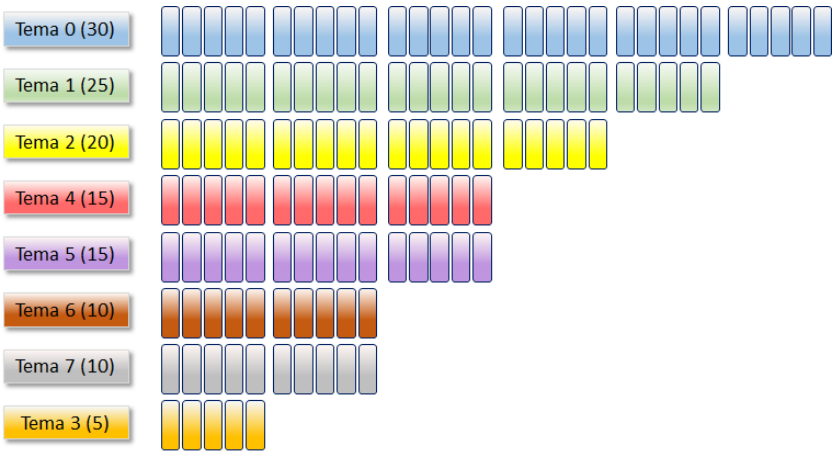
## Exemple execució

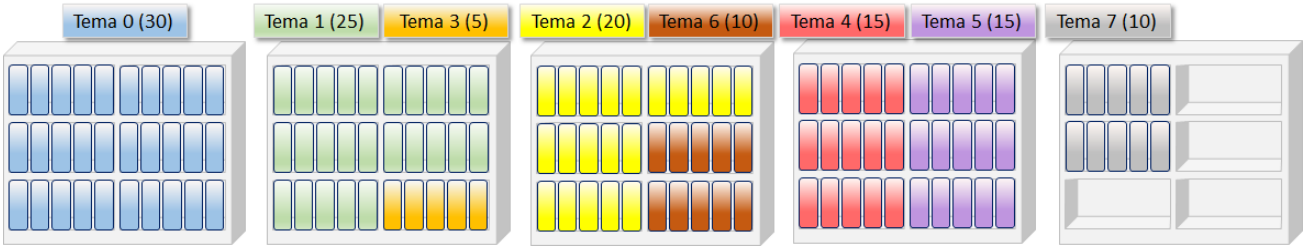
```
Indica la capacitat que tenen les prestatgeries que tens
30
Quantes tematiques diferents vols ubicar?
8
Dades entrades:
**********
Nom de la tematica tema 2 amb: 20 llibres
Nom de la tematica tema 1 amb: 25 llibres
Nom de la tematica tema 0 amb: 30 llibres
Nom de la tematica tema 5 amb: 15 llibres
Nom de la tematica tema 4 amb: 15 llibres
Nom de la tematica tema 3 amb: 5 llibres
Nom de la tematica tema 6 amb: 10 llibres
Nom de la tematica tema 7 amb: 10 llibres

Necessitem 5 prestatgeries
Distribucio:
Prestatgeria 0 hi ubiquem:
Nom de la tematica tema 0 amb: 30 llibres

Prestatgeria 1 hi ubiquem:
Nom de la tematica tema 1 amb: 25 llibres
Nom de la tematica tema 3 amb: 5 llibres

Prestatgeria 2 hi ubiquem:
Nom de la tematica tema 2 amb: 20 llibres
Nom de la tematica tema 6 amb: 10 llibres

Prestatgeria 3 hi ubiquem:
Nom de la tematica tema 4 amb: 15 llibres
Nom de la tematica tema 5 amb: 15 llibres

Prestatgeria 4 hi ubiquem:
Nom de la tematica tema 7 amb: 10 llibres


Tematiques no ubicades:
Totes estan ubicades
```

# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

## Pla d'execució

1. Ordenar candidats de grans a petits
2. Recorregut dels candidats
3. Selecció prestatgeria
   Tot el temari ha d'anar junt
   Aprofitar l'espai restant, seleccionar el que menys espai deixi
   Si no hi cap, prestatge nou
4. Col·locar al prestatge




Animació

https://www.youtube.com/watch?v=qfTCWoniqlo

# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

## Respostes

1. (0.5 punts) Raona si la funció de selecció indicada trobarà o no la millor solució al problema. Justifica la resposta.

   Sempre trobarà la millor solució al problema. El fet de tractar les temàtiques de més gran a més petita i assignar-li la prestatgeria que deixa el menor espai lliure garanteix de trobar sempre la millor solució, per tant no resoldríem aquest exercici amb bactracking ja que trigaria més per trobar la mateixa solució.

3. (1 punt) Aquest problema també pot esser resolt aplicant la tècnica del backtracking. Fes l'anàlisi d'aquest problema contestant totes les preguntes indicades:

   Per què l'esquema de backtracking és aplicable per a resoldre aquest enunciat?

   La solució pot ésser expressada com una **seqüència de decisions**, per tant es pot aplicar la tècnica del backtracking per trobar la millor solució.

   Com hi poden haver temàtiques que no hi caben en una prestatgeria, prèviament a la invocació del backtracking eliminaríem aquelles temàtiques de capacitat superior a la de les prestatgeries. Així totes les temàtiques s'han d'ubicar.

   Determina quines decisions ha de prendre el programa.

   La decisió a prendre en cada nivell de l'arbre, corresponent a l'espai de cerca, és "**aquesta temàtica T on la ubico?**", s'ha decidir per a cadascuna de les temàtiques per tant la solució vindrà donada quan arribem a una fulla de l'arbre.
   **Acceptable**: Una decisió serà acceptable si tots els llibres de la temàtica caben a la prestatgeria indicada.

   Quin és el criteri per determinar si un conjunt de decisions és o no solució.

   **Solució**: Serà solució quan arribem a una fulla.

   Quin és el criteri per determinar si un conjunt de decisions és o no completable.

   **Completable**: mentre no hi arribem a una fulla.

   Dibuixa l'espai de cerca del problema, és a dir, l'arbre que recorrerà la tècnica del backtracking, especificant quina serà l'alçada i l'amplada, indicant si són valors exactes o valors màxims.

   **Espai cerca**: l'alçada exacta corresponent al nombre de temàtiques de mida no superior a la cabuda de la prestatgeria i amplada màxima, exactament idèntica a la alçada, en el pitjor dels casos hi haurà una temàtica en cada prestatgeria.

   **Optimitzar**: Quan es troba una solució s'ha de comparar el nombre de prestatgeries usades i en cas de ser menor a la de la millor solució agafar-la com a millor.

   Amb el teu plantejament cal usar la tècnica del marcatge?

   No cal usar marcatge doncs en una prestatgeria s'hi pot ubicar més d'una temàtica.

# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

## Implementació

```
import java.util.Arrays;


public class Aplicacio {

    private Tematica[] candidats;
    //2.1 Afegir atributs per emmagatzemar la solució trobada
    private Prestatgeria[] prestatgeries;
    private int quantes;
    private String noUbicats;

    public Aplicacio(int quants) {
        candidats = new Tematica[quants];
        //2.2 Crear i inicialitzar atributs afegits
        prestatgeries = new Prestatgeria[quants]; // cas pitjor: una per cads tema
        quantes = 0;
        noUbicats="";
        omplenaDades();
    }


    public static void main(String args[]) {
        System.out.println("Indica la capacitat que tenen les prestatgeries que tens");
        Prestatgeria.capacitatTotal = 30; //Keyboard.readInt();
        System.out.println("Quantes tematiques diferents vols ubicar? ");
        int quants = 8; //Keyboard.readInt();
        // 2.3 Crear objecte Aplicacio, invoca al mètode voraç i mostra la solució trobada
        Aplicacio m = new Aplicacio(quants);
        m.solucio();

        System.out.println("\nNecessitem " + m.quantes + " prestatgeries");
        System.out.println("Distribucio:\n"+ m);
        System.out.println("Tematiques no ubicades: ");
        if (m.noUbicats!="")System.out.println("\n"+ m.noUbicats);
        else System.out.println("Totes estan ubicades");
    }


    private void omplenaDades() {
        candidats[0] = new Tematica("tema 2", 20);
        candidats[1] = new Tematica("tema 1", 25);
        candidats[2] = new Tematica("tema 0", 30);
        candidats[3] = new Tematica("tema 5", 15);
        candidats[4] = new Tematica("tema 4", 15);
        candidats[5] = new Tematica("tema 3", 5);
        candidats[6] = new Tematica("tema 6", 10);
        candidats[7] = new Tematica("tema 7", 10);
    }



    public void solucio() {
/* 2.4 Implementació mètode voraç. Decidiu vosaltres el retorn i paràmetres.
* En la seva execució és imprescindible que s'invoqui al mètode següent corresponent
* a la implementació de la funció de selecció, el que determina el següent candidat
* a considerar.
*/
        // ordena de menys a mes capacitat les tematiques
        Arrays.sort(candidats);
        // recorregut invers de tots els candidats
        for (int i = candidats.length - 1; i >= 0; i--) {
            int quina = funcioSeleccio(candidats[i]);
            switch (quina) {
                case -1: // no la ubiquem
```

```
                    noUbicats += candidats[i].getNom() + "\n";
                    break;
                case -2: // estanteria nova
                    prestatgeries[quantes] = new Prestatgeria();
                    prestatgeries[quantes].addTematica(candidats[i]);
                    prestatgeries[quantes].ocuparEspai(candidats[i].getNumLlibres());
                    quantes++;
                    break;
                default: // ho possem a quina
                    prestatgeries[quina].addTematica(candidats[i]);
                    prestatgeries[quina].ocuparEspai(candidats[i].getNumLlibres());
                    break;
            }
        }
    }


private int funcioSeleccio(Tematica p) {
/* 2.5 Implementació del mètode privat que aplica la funció de
* selecció indicada en l'enunciat. Determineu vosaltres el
* retorn i els paràmetres necessaris.
*/

    // buscar la prestatgeria on hi cap i sobre menys espai
    if (p.getNumLlibres() > Prestatgeria.capacitatTotal) return -1;//no hi cap
    int quina = -1;
    int espaiSobrant = Prestatgeria.capacitatTotal;
    // recorregut per les prestatgeries existents per buscar espai
    for (int j = 0; j < quantes; j++) {
        int espai = prestatgeries[j].getEspaiOcupat() + p.getNumLlibres();
        if (espai <= Prestatgeria.capacitatTotal) {
            // comparar l'espai sobrant entre les prestatgeries existents
            if (Prestatgeria.capacitatTotal - espai < espaiSobrant) {
                quina = j;
                espaiSobrant = Prestatgeria.capacitatTotal - espai;
            }
        }
    }
    if (quina == -1) return -2; // una nova
    else return quina;
}

    public String toString() {
/* 2.6 Ha de crear i retornar una cadena amb la solució trobada.
* Ha d'indicar el número de prestatgeries necessàries, per a cadascuna d'elles les
* temàtiques que allotja i finalment les temàtiques que no es poden ubicar
* i per tant no es posen a la venda.
*/

        String re = "";
        for (int i = 0; i < quantes; i++) {
            re += "Prestatgeria "+ i +" hi ubiquem: \n"
            + prestatgeries[i].toString() +"\n";
        }
        return re;
    }
}
```

# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

```java
import java.util.ArrayList;
import java.util.List;

public class Prestatgeria {
    public static int capacitatTotal;
    private int espaiOcupat;
    private List<Tematica> contingut;

    public Prestatgeria() {
        espaiOcupat = 0;
        contingut = new ArrayList<Tematica>();
    }

    public void addTematica(String tematica, int mida) {
        this.addTematica(new Tematica(tematica, mida));
    }

    public void addTematica(Tematica p) {
        contingut.add(p);
    }

    public int getEspaiOcupat() {
        return espaiOcupat;
    }

    public void ocuparEspai(int espai) {
        espaiOcupat += espai;
    }

    public String toString() {
        String r = "";
        for (int i = 0; i < contingut.size(); i++) {
            r += contingut.get(i) + "\n";
        }
        return r;
    }
}
```

# Exercici 2 - Tècnica Voraç Examen 16 Des 2016

```
public class Tematica implements Comparable {
    private String nom;
    private int numLlibres;

    public Tematica(String nom, int num) {
        this.nom = nom;
        this.numLlibres = num;
    }

    public String getNom() {
        return nom;
    }

    public int getNumLlibres() {
        return numLlibres;
    }

    public int compareTo(Object o) {
        Tematica t = (Tematica) o;
        if (numLlibres < t.numLlibres) return -1;
        if (numLlibres == t.numLlibres) return 0;
        return 1;
    }

    public String toString() {
        return "Nom de la tematica " + nom + " amb: " + numLlibres;
    }
}
```