



Tècnica de backtracking

Exercici backtracking

Assignació d'àrbitres a partits
de futbol

Exercici 13 pàgina 55



Tècnica de backtracking

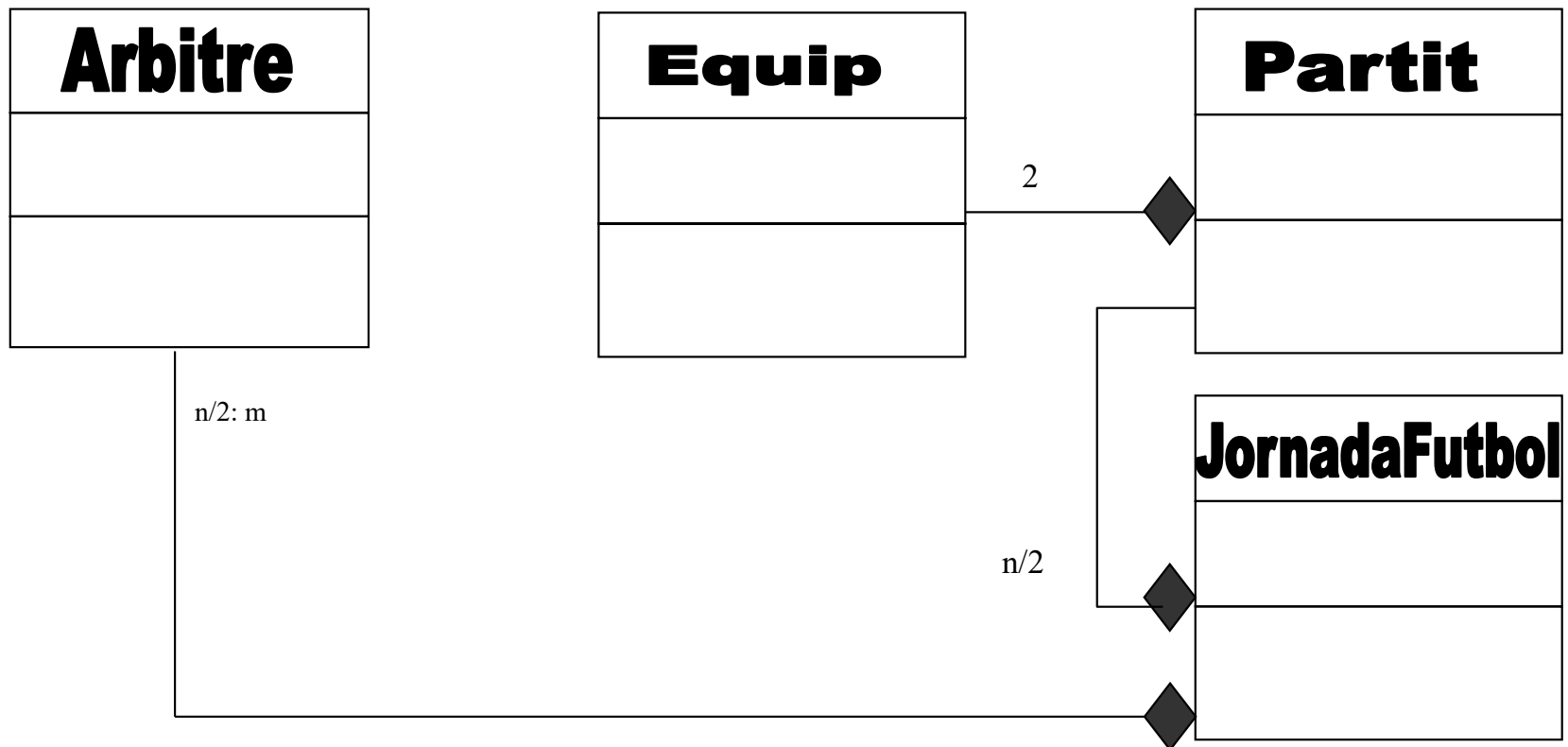
Assignació Àrbitres

- **Enunciat:** Assignació d'àrbitres als partits d'una jornada futbolística. Dades:
 - N equips (parell) → **N/2 partits**
 - **M àrbitres** → $M \geq N/2$
 - Valoració equip a cada àrbitre: $[0,10]$
- **L'objectiu** és, per una jornada determinada, assignar un àrbitre **diferent** a cada partit, de manera que, la puntuació total dels àrbitres assignats sigui **màxima** tenint en compte les **preferències** dels equips. Cal aplicar la tècnica del backtracking per a trobar la **millor** assignació.
- Cal denotar que **no serà admissible** com a solució, assignar un àrbitre a un determinat partit si l'equip local i/o l'equip visitant són de la mateixa comarca que l'àrbitre.
Restricció del problema.



Tècnica de backtracking

- Disseny de classes:



Assignació Arbitres



Tècnica de backtracking

Assignació Arbitres

```
public class Arbitre{
    private String nom;
    private String comarca;
    public Arbitre(String nom, String comarca){
        this.nom=nom;
        this.comarca=comarca;
    }
    public String getNom(){return nom;}
    public String getComarca(){return comarca;}
    public String toString(){
        return ("Nom Arbitre: " + nom + " Comarca " +
                comarca + "\n");
    }
} //fi classe
```



Tècnica de backtracking

Assignació Arbitres

```
public class Equip{
    private String nom;
    private String comarca;
    public Equip(String nom, String comarca){
        this.nom=nom;
        this.comarca=comarca;
    }
    public String getNom(){return nom;};
    public String getComarca(){return comarca;}
    public String toString(){
        return ("Nom: "+ nom + "
                Comarca " + comarca + "\n");
    }
}
```



Tècnica de backtracking

Assignació Arbitres

```
public class Partit{
    private Equip Local;
    private Equip Visitant;
    private int golsLocal;private int golsVisitant;
    public Partit(Equip local, Equip visitant){
        Local=local; Visitant=visitant;
        golsLocal=0; golsVisitant=0;
    }
    public Equip getLocal(){return Local;}
    public Equip getVisitant (){return Visitant;}
    public void setGolsLocal(int valor){golsLocal=valor;}
    public void setGolsVisitant(int valor){
        golsVisitant=valor;}
    public String toString(){
        return ("Local "+ Local.toString() + " Visitant " +
            Visitant.toString() +"\n"); }
}
```



Tècnica de backtracking

Assignació Arbitres

```
public class JornadaFutbol{
    private int numJornada;
    private Arbitre []Arbitres;
    private int numArbitres; //dim real de la taula prèvia
    private Partit []Partits;
    private int numPartits; //dim real de la taula prèvia
    private int [][]preferencies; //files→equips -
    //columnes→àrbitres

    private class Parella{ //classe privada
        Arbitre a; Partit p;
        public Parella(Arbitre a, Partit p)
        {this.a=a; this.p=p;}
    }

    private Parella MillorSolucio[]; private int costMillor;
    private void omplenaPreferencies(){
        /*sentències Random per omplenar la taula preferencies*/}
    private void omplenaPartits(){
        /*omplena la taula Partits amb els partits de la jornada*/ }
}
```

**Enunciat
indica COM
s'emmagatzem
les dades en
les
preferències**

Entrada de dades



Tècnica de backtracking

Inició Arbitres

```
public class JornadaFutbol{  
    private int numJornada;  
    private Arbitre []Arbitres;  
    private int numArbitres; //dim real de la taula prèvia  
    private Partit []Partits;  
    private int numPartits; //dim real de la taula prèvia  
    private int [][]preferencies; //files→equips -  
    columnes→àrbitres  
    private class Parella{ //o  
        Arbitre a; Partit p;  
        public Parella(Arbitre  
            {this.a=a; this.p=p;  
    }  
}
```

