

**Grup**  
Enric Viñas Mas  
Carles Rigat Almansa

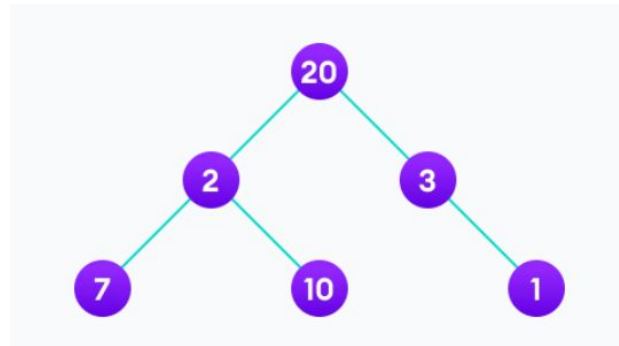
# *Algoritme Greedy Best First Search*

17 de maig de 2022

# 1. Explicació

## Algoritme Greedy / Voraç - Cerca Informada - Millora del BFS

Busca sempre el node més òptim de manera local, però no global



El més òptim seria  $20 + \mathbf{2} + 10$ , però el **Greedy** va  $20 + \mathbf{3} + 1$

Es poden trobar similituds amb el Hill-Climbing

# PROS

- ✓ És **fàcil** d'entendre
- ✓ Consumeix **pocs recursos**
- ✓ Fàcil d'implementar
- ✓ Al ser un algoritme de cerca informada, és més **eficient**, té millor **rendiment** i **cost** més **baix** (que un algoritme de cerca no informada)

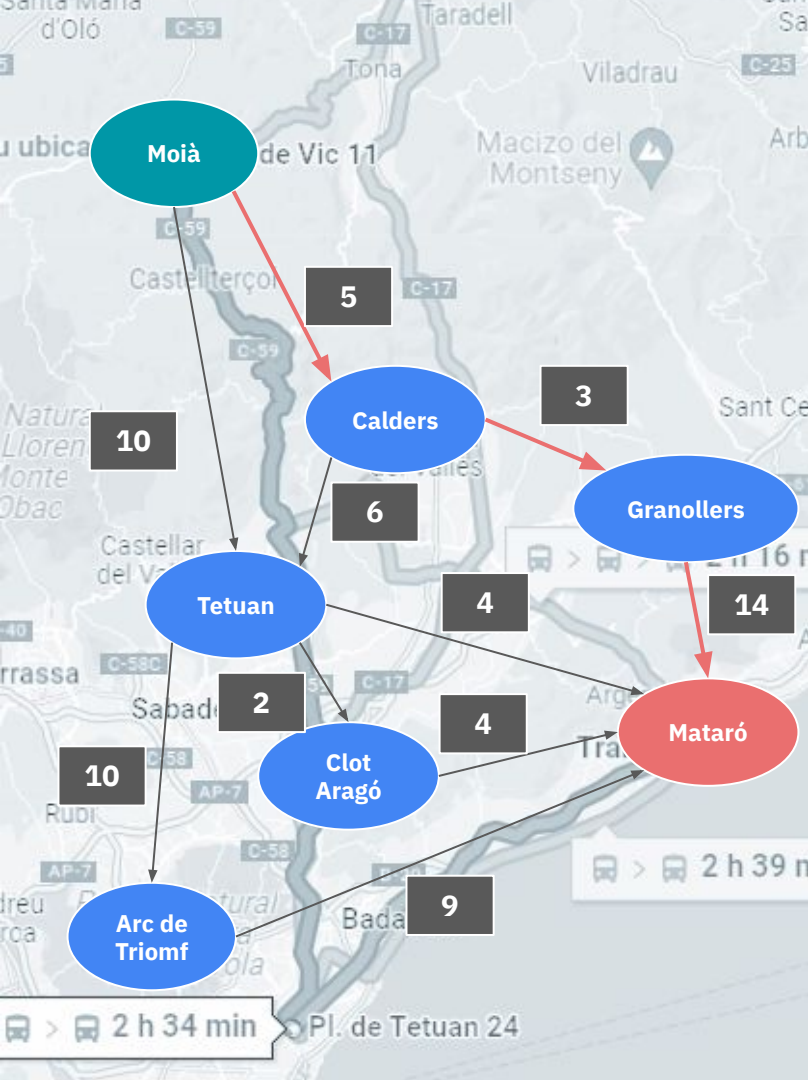
# CONTRES

- ✗ **No sempre** troba la solució **òptima global**
- ✗ Al ser un algoritme de **cerca informada**, requerim tenir coneixement concret sobre característiques del problema



## ***2. Exemple***

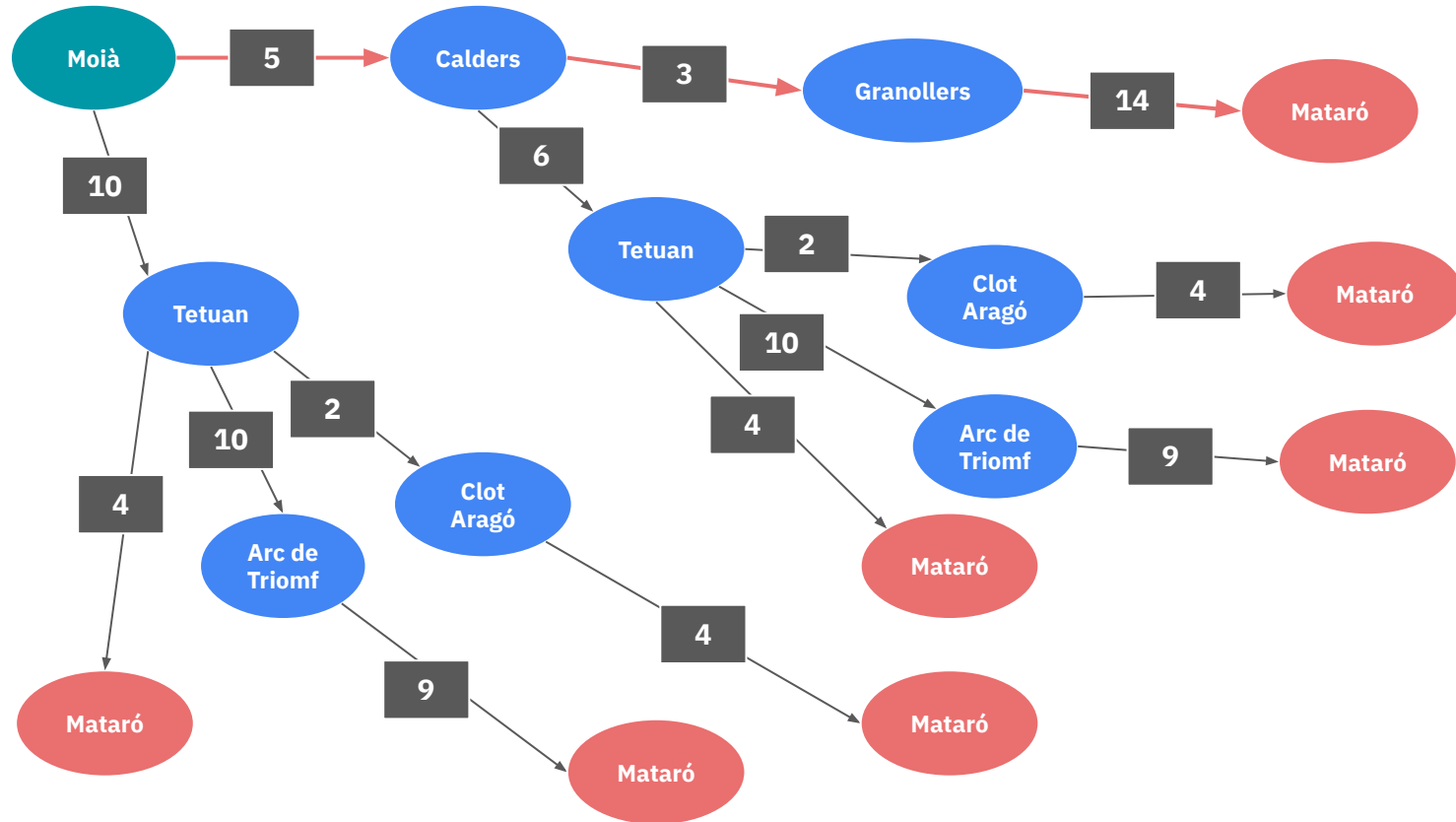
***Trajecte  
Moià - Mataró***



## MOIÀ (inici)

- Tetuan
  - Clot Aragó
    - **Mataró 16**
  - Arc de triomf
    - **Mataró 29**
  - **Mataró 14**
- Calders
  - Granollers
    - **Mataró 22**
  - Tetuan
    - **Mataró 15**

## ***Versió horitzontal***



# 3. Implementació en Python

```
Greedy():
global actual
global possiblePath
global finalPath

while(actual != desti):
    possiblePath.clear()

    for key, cost in actual.children.items():

        if(checkValidity(key)):
            possiblePath[key] = cost
            print("Branca analitzada\n")

    print("\nEls possibles camins son per ", possiblePath, "\n\n")

    actualSet = selectLessCost(possiblePath)

    AddCost(actualSet)

    print("Següent parada, ", actual.name)

    finalPath.append(actual)
```

```
4 > class Tree:
16
17 #Relacionem els nodes i assignem valors
18 > def createTree():
28
29
30 > def checkValidity(node):
56
57 #Basicament mirem el diccionari on guardem nodes i el seu cost
58 # i agafem el que te un cost menor
59 # IMPORTANT: es fa despres de comprovar que es viable arribar al desti
60 > def selectLessCost(options):
72
73 #Funcio per sumar el cost del viatge
74 > def AddCost(actualSet):
81
82
83 > def Greedy():
106
107
108
109 # Variables globals
110 moia = Tree("Moia")
111 tetuan = Tree("Tetuan")
112 calders = Tree("Calders")
113 clot_arago = Tree("Clot aragó")
114 arc_triomf = Tree("Arc del triomf")
115 granollers = Tree("Granollers")
116 mataro = Tree("Mataró")
117 mataro.leaf = True
118
119 actual = moia
120 desti = mataro
121 cost_a = 0
122 possiblePath = {}
123 finalPath = []
124 finalPath.append(moia)
125
126 print("Estudiem el recorregut entre ", actual.name, " i ", desti.name, "\n")
127
128 createTree()
129
130 Greedy()
131
132 print("-----\n")
133 print("El recorregut final (amb algoritme Greedy) passa per: ", finalPath, "\n")
134 print("El cost del recorregut total es: ", cost_a)
135 print("-----\n")
```

# 3. Implementació en Python

```
PS G:\Mi unidad\evinas@eturo.org 2018-12-17 17 12\Tecnocampus\3  
inas@eturo.org 2018-12-17 17 12\Tecnocampus\3er\3er trimestre/T  
Estudiem el recorregut entre Moia i Mataró
```

```
Studing Node: Calders  
  Next child: Granollers  
Studing Node: Granollers  
  Reached Mataró !!!  
  Next child: Tetuan  
Studing Node: Tetuan  
  Next child: Clot aragó  
Studing Node: Clot aragó  
  Reached Mataró !!!  
  Next child: Arc del triomf  
Studing Node: Arc del triomf  
  Reached Mataró !!!  
  Reached Mataró !!!  
Branca analitzada
```

```
Studing Node: Tetuan  
  Next child: Clot aragó  
Studing Node: Clot aragó  
  Reached Mataró !!!  
  Next child: Arc del triomf  
Studing Node: Arc del triomf  
  Reached Mataró !!!  
  Reached Mataró !!!  
Branca analitzada
```

```
Els possibles camins son per {Calders: 5, Tetuan: 10}
```

```
Següent parada, Calders  
Studing Node: Granollers  
  Reached Mataró !!!  
Branca analitzada  
  
Studing Node: Tetuan  
  Next child: Clot aragó  
Studing Node: Clot aragó  
  Reached Mataró !!!  
  Next child: Arc del triomf  
Studing Node: Arc del triomf  
  Reached Mataró !!!  
  Reached Mataró !!!  
Branca analitzada
```

```
Els possibles camins son per {Granollers: 3, Tetuan: 6}
```

```
Següent parada, Granollers  
Studing Node: Mataró  
Branca analitzada
```

```
Els possibles camins son per {Mataró: 14}
```

```
Següent parada, Mataró  
-----
```

```
El recorregut final (amb algoritme Greedy) passa per: [Moia, Calders, Granollers, Mataró]
```

```
El cost del recorregut total es: 22  
-----
```



# ***Bibliografia***

- Greedy Algorithm. (2022). Retrieved 12 May 2022, from <https://www.programiz.com/dsa/greedy-algorithm#:~:text=A%20greedy%20algorithm%20is%20an,if%20the%20choice%20is%20wrong>.
- Algorisme voraç - Viquipèdia, l'enciclopèdia lliure. (2022). Retrieved 12 May 2022, from [https://ca.wikipedia.org/wiki/Algorisme\\_vora%C3%A7](https://ca.wikipedia.org/wiki/Algorisme_vora%C3%A7)
- Greedy Algorithms - GeeksforGeeks. (2022). Retrieved 12 May 2022, from <https://www.geeksforgeeks.org/greedy-algorithms/>