

Jocs de Proves del Moodle

Posarem a prova el programa amb tots els jocs de proves proporcionats pels professors, que estan disponibles al Moodle.

aixetasATancar

```
≡ fitxer_sortida
1 tancar
2 c12
3 |
```

arbres1

```
≡ fitxer_sortida
1 x1 es un arbre
2 x2 no es un arbre
3 x3 no es un arbre
4 |
```

backtrackCabalMinim

```
≡ fitxer_sortida
1 cabal minim
2 50.0
3 |
```

cabalMinim2

```
cabal minim
350.0
cabal minim
175.0
cabal minim
200.0
cabal minim
100.0
cabal minim
0.0
cabal minim
0.0
cabal minim
0.0
cabal minim
0.0
```

cicles1

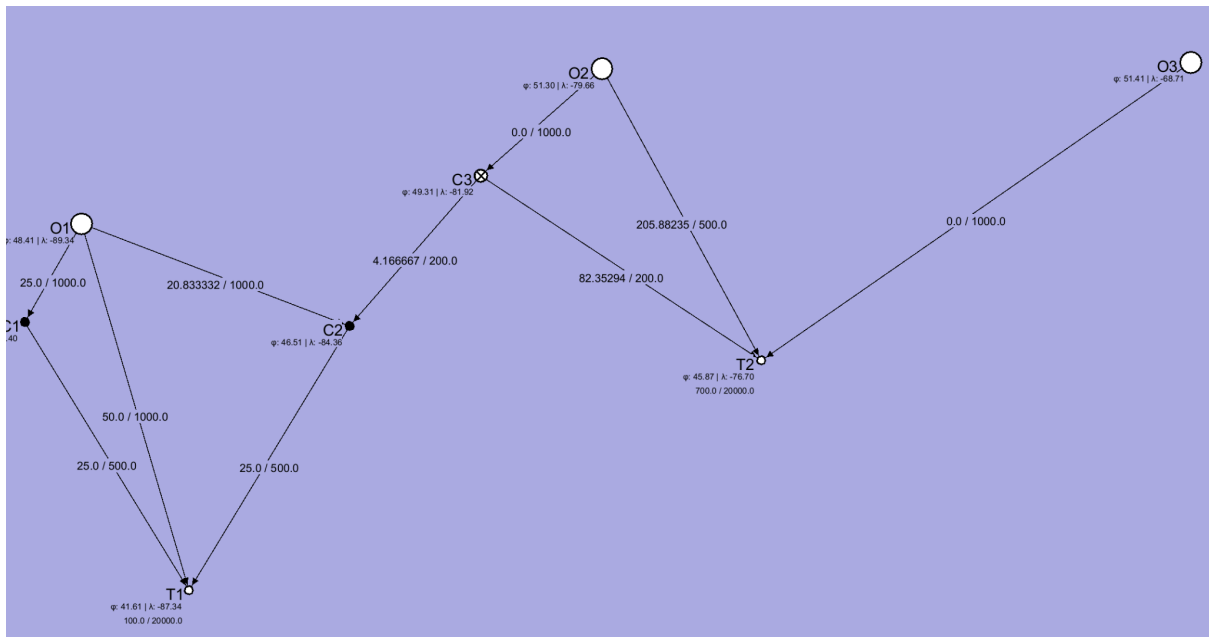
```
≡ salida.txt
1 03 te cicles
2 Error: opció dibuix - La xarxa no hauria de tenir cicles
3 |
```

Indica bé que té cicles, i llavors quan li demana dibuixar, salta un error. Com que té cicles, no pot fer el dibuix. En l'enunciat de la pràctica s'especifica que en aquesta opció hem de suposar que la xarxa no té cicles.

complet1

```
1 01 no te cicles
2 01 es un arbre
3 cabal minim
4 50.0
5 exces cabal
6 C12-T12
7 exces cabal
8 C11-C12
9 exces cabal
10 04 no te cicles
11 04 no es un arbre
12 exces cabal
13 C12-T12
14 |
```

dibuix1



errors1

```
1 | Error: ja existeix a les xarxes una aixeta amb nom o1.
2 |
```

errors2

```
1 | Error: For input string: "connectar"
2 |
```

errors3

```
1 | Error: Coordenada errònia - 50:12:100S,155:12:12W
2 |
```

proximitat1

```
1 | proximitat
2 | O1
3 | C1
4 | C2
5 | C3
6 | O2
7 | T1
8 | T2
9 | O3
10 |
```

Com es pot observar, ens ha donat els mateixos resultats. Cal destacar que els missatges d'error no són els mateixos que els exemples de sortida penjats al Moodle.

Jocs de proves nostres

MAXFLOW

En primer lloc, hem construït el graf amb un superorigen i un superterminal. Bàsicament, connectem el superorigen amb les connexions que té cada origen, i si dos orígens van a una mateixa connexió, això serien dues canonades en el graf original, però el que fem és sumar les capacitats i així només hi haurà una canonada amb el superorigen. De la mateixa manera amb el superterminal. Per entendre millor aquest procés mostrarem el graf amb el mètode dibuixar() i després amb fluxMaxim():

terminal

T11

41:57:47.29N,2:49:53.64E

30

terminal

T12

41:58:24.45N,2:48:52.3E

50

connexio

C11

41:57:47.34N,2:49:53.85E

connexio

C12

41:57:47.34N,2:49:53.85E

origen

O1

41:53:7.56N,2:33:14.32E

connectar

C11

T11

250

connectar

C11

C12

250

connectar

C12

T12

20

connectar

O1

C11

250

origen

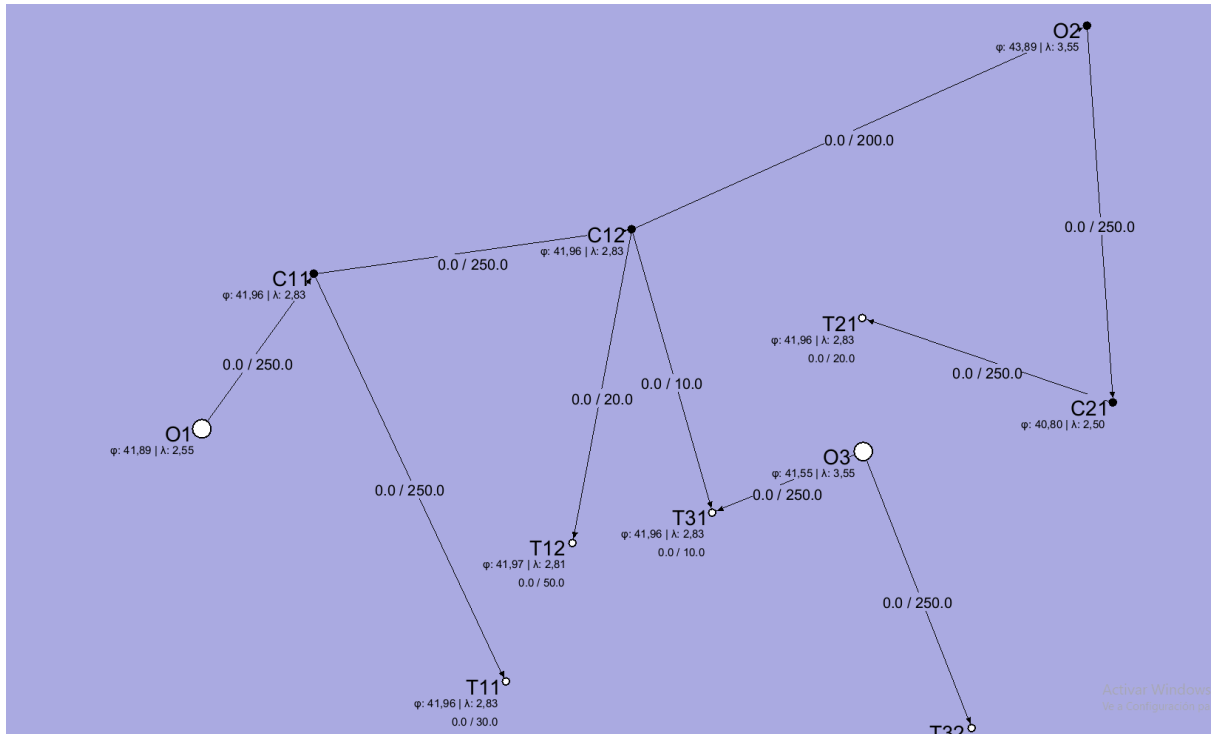
O2

43:53:7.56N,3:33:14.32E

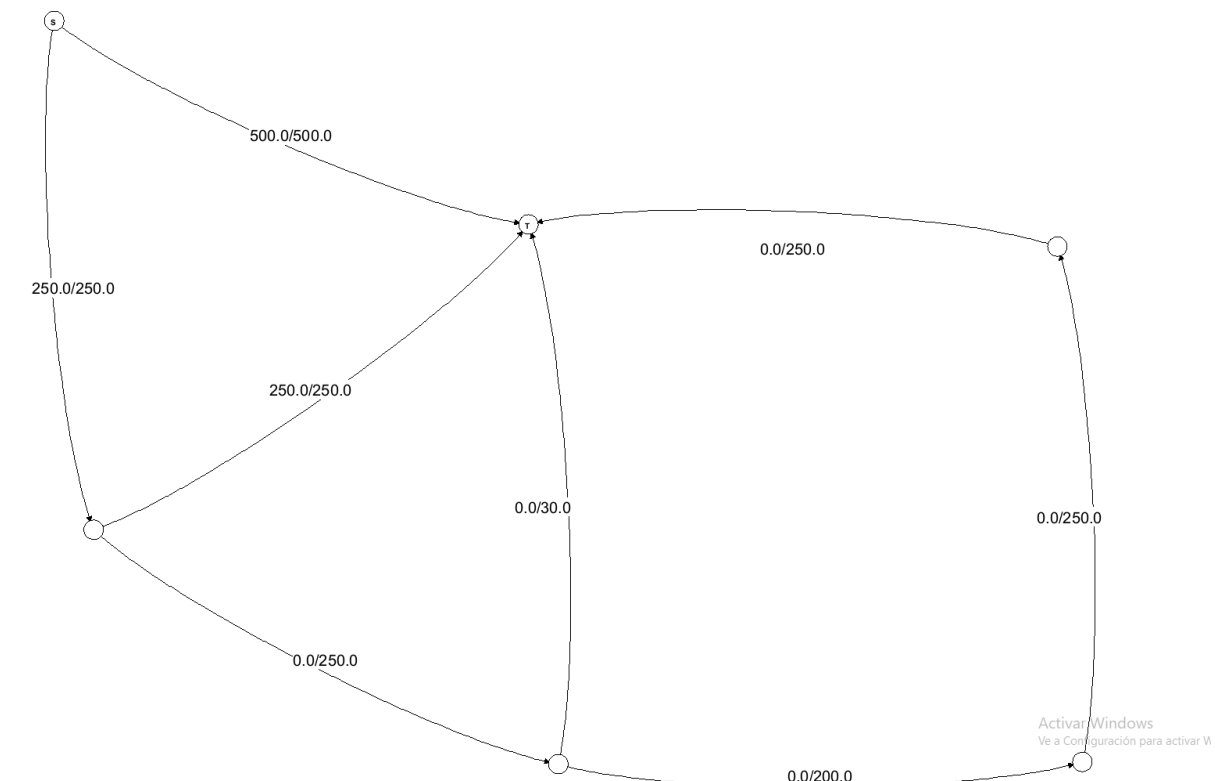
connectar
C12
O2
200
origen
O3
41:33:6.56N,3:33:14.32E
origen
O4
41:57:47.29N,2:49:53.64E
terminal
T21
41:57:47.29N,2:49:53.64E
20
terminal
T31
41:57:47.29N,2:49:53.64E
10
terminal
T32
41:57:47.29N,2:49:53.64E
10
terminal
T41
41:57:47.29N,2:49:53.64E
50
connectar
C12
T31
10
connexio
C21
40:47:47.24N,2:29:43.35E
connectar
O2
C21
250
connectar
C21
T21
250
connectar
O3
T31
250
connectar
O3
T32
250
connectar
O4
T41

250
dibuix
O3

Sortida:



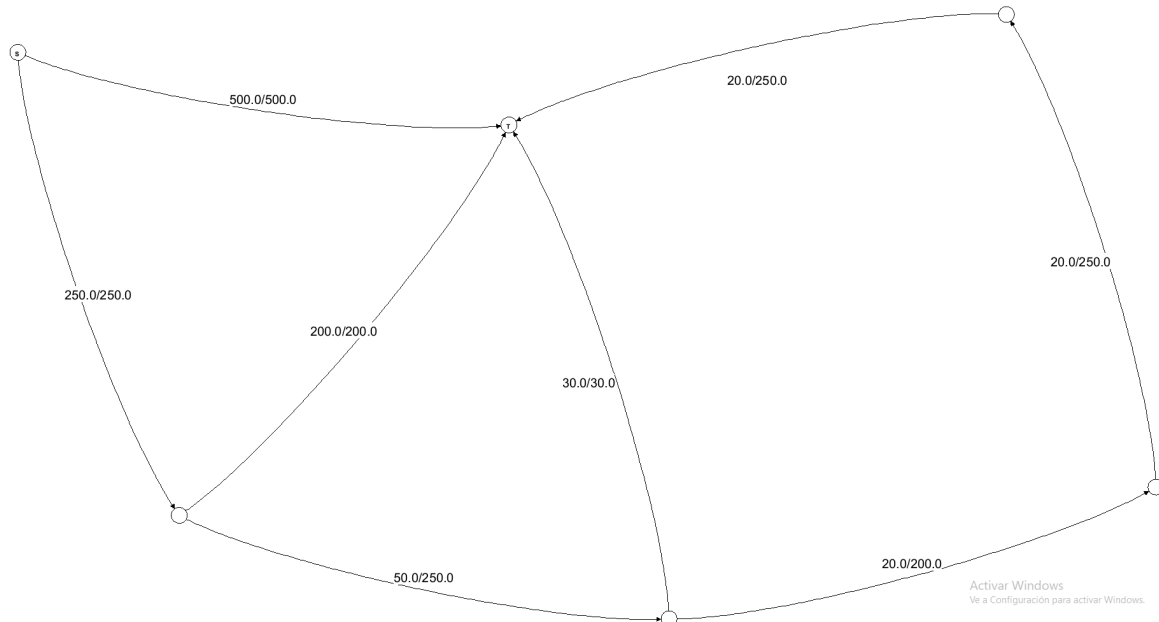
Ara canviem la penúltima línia d'entrada per max-flow.
maxFlow1



Com podem veure, les capacitats de O3-T31 i O3-T32 queden sumades en una sola canonada.

En aquest exemple podem veure el correcte funcionament del problema, però hi ha un segon exemple que reparteix les capacitats de les canonades per veure millor el funcionament al complet.

maxFlow2



En aquest segon exemple podem veure que mentre hi ha camí es distribuirà flux per tota la xarxa respectant les capacitats i fent una bona gestió del flux residual.

creacio

Creem una xarxa i després la dibuixem per tal de veure si l'ha creat correctament.

origen

O1

41:53:7.56N,2:33:14.32E

connexio

C1

21:53:7.56N,2:33:14.32E

connectar

O1

C1

27.45

terminal

T1

1:58:24.45N,2:48:52.3E

3456.23

connectar

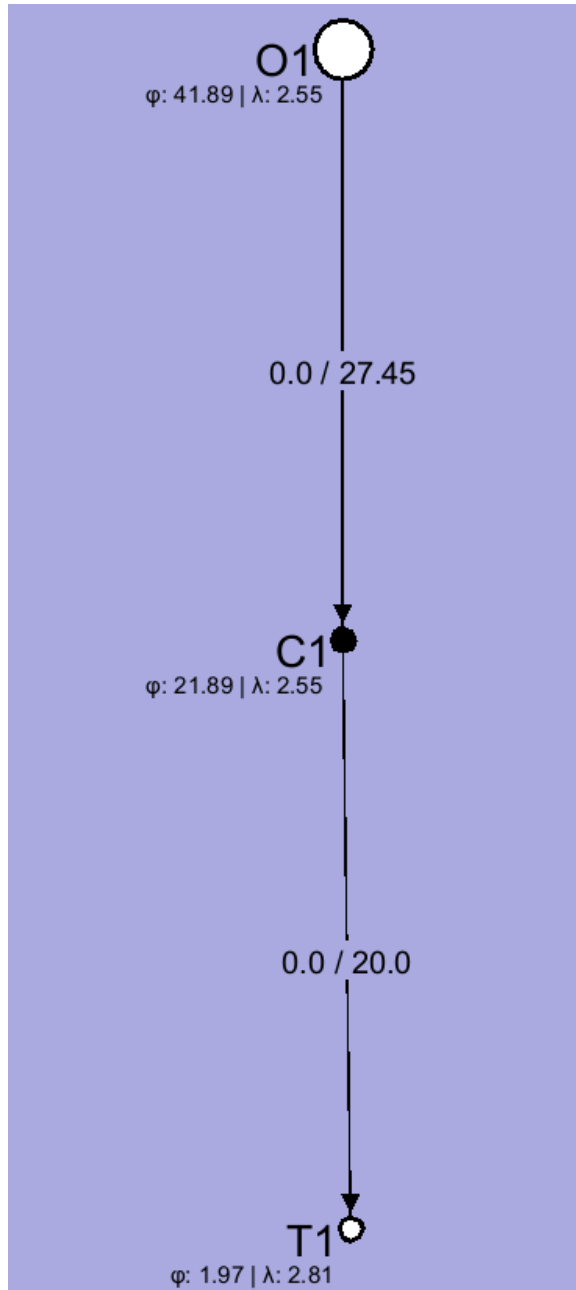
C1

T1

20

dibuix

O1



Com es pot observar el resultat és correcte.

Backtrack

Ara posarem a prova el backtracking amb la xarxa creada del fitxer de prova creació. Per tant, els següents fitxers de prova comencen tots amb la mateixa entrada del fitxer de prova creació.

backtrack1

Posem a prova què succeeix si intentem desfer més operacions d'obrir i tancar aixetes de les que s'han realitzat. En aquest cas, mirem de desfer 50 operacions, quan no se n'han produït cap.

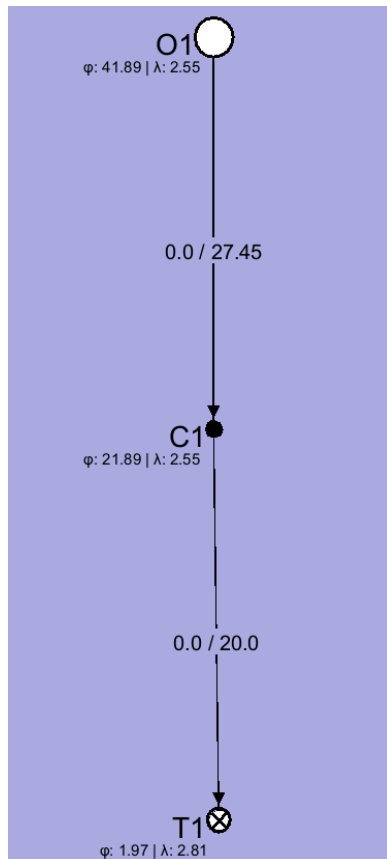
```
backtrack
50
```

No succeeix res, ja que no hi ha operacions per poder desfer. La sortida de text és buida.

backtrack2

Comprovem que tingui en compte les operacions tot i que no canviïn l'estat de les aixetes.

```
tancar
T1
obrir
O1
backtrack
1
dibuix
O1
```



Com es pot observar, tot i que l'última operació d'obrir aixeta no canvia l'estat de l'aixeta, sí que es té en compte a l'hora de fer el backtracking.

backtrack3

Ara, en comptes de desfer només un pas, desfarem dos passos enrere. Això ens hauria de fer retornar a l'estat original de la xarxa.

tancar

T1

obrir

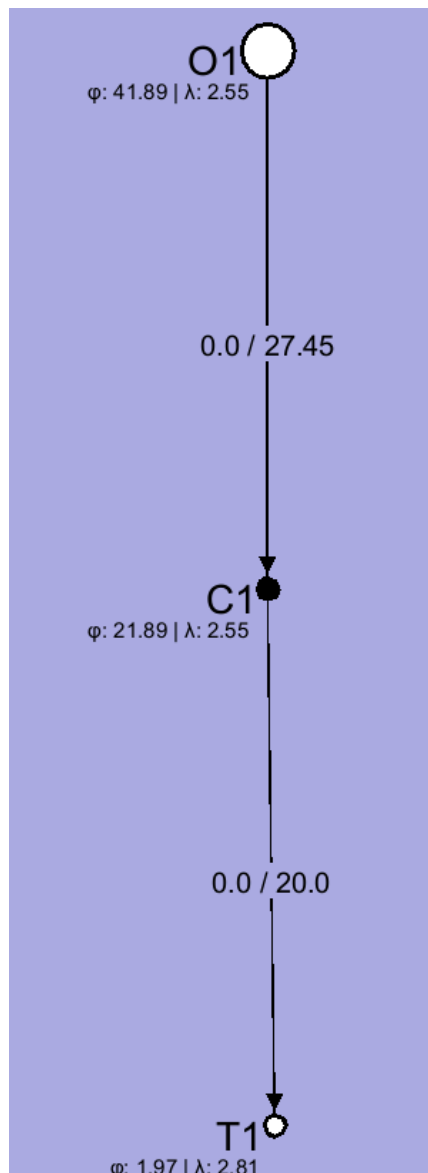
O1

backtrack

2

dibuix

O1



Tal com havíem dit, tornem a l'estat original.

Afegir

Posarem a prova els mètodes que afegeixen un nou node a la xarxa. Sobretot comprovarem que no es pugui afegir un node amb el mateix identificador. Tots els fitxers de prova comencen amb la xarxa creada amb el fitxer creació.

afegir1

origen

O1
60:53:7.56N,2:33:14.32E

```
1 Error: ja existeix a les xarxes una aixeta amb nom O1.  
2
```

afegir2

connexio
O1
60:53:7.56N,2:33:14.32E

```
1 Error: ja existeix a les xarxes una aixeta amb nom O1.  
2
```

afegir3

terminal
O1
60:53:7.56N,2:33:14.32E
3456.23

```
1 Error: ja existeix a les xarxes una aixeta amb nom O1.  
2
```

Com es pot observar, tant si s'intenta afegir origen, terminal o connexió, es té en compte si l'identificador ja existeix a la xarxa, i si és així, salta un error.

Connectar

Tots els fitxers de prova comencen amb la xarxa creada amb el fitxer creació.
Mirarem de posar a prova el mètode que connecta dos nodes.

connectar1

Què passa si intentem connectar dos nodes que no existeixen?

connectar
O2
T2
60

```
1 | Error: El primer node no pertany a la xarxa
2
```

connectar2

Què passa si el primer node no existeix?

```
connectar
O2
T1
60
```

```
1 | Error: El primer node no pertany a la xarxa
2
```

connectar3

Què passa si el segon node no existeix?

```
connectar
O1
T2
60
```

```
1 | Error: El segon node no pertany a la xarxa
2
```

connectar4

Què passa si connectem dos nodes que ja estan connectats?

```
connectar
C1
O1
60
```

```
1 | Error: Ja estan connectats
2
```

connectar5

Què passa si el primer node és terminal?

```
origen
```

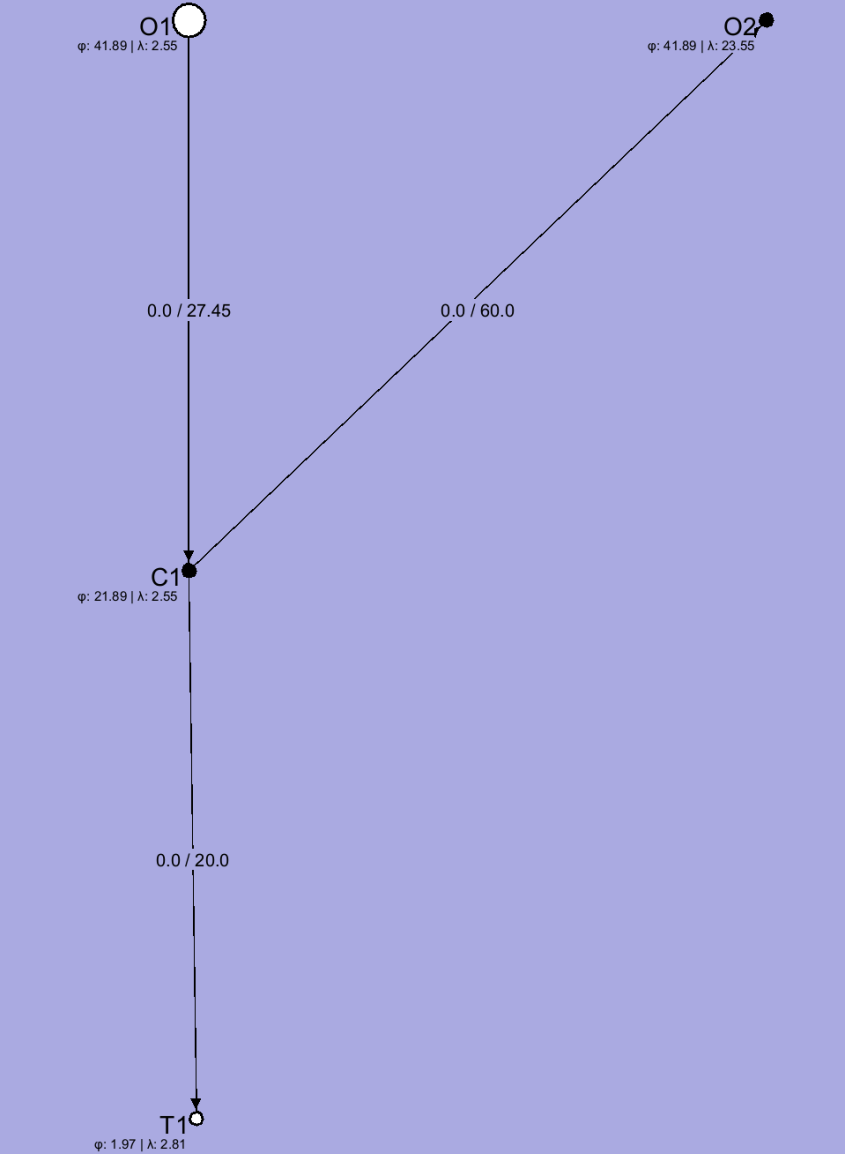
O2
41:53:7.56N,60:33:14.32E
connectar
T1
O2
60

```
1 Error: El primer node es un punt terminal
2
```

connectar6

Què passa si el segon node és origen?

O2
41:53:7.56N,60:33:14.32E
connectar
C1
O2
60
dibuix
O1



El node O2 es converteix d'origen a connexió.

Establir cabal

cabal

Mirarem d'establir cabal a un punt que no és origen.

cabal

C1

250

```
1 Error: C1 no és un origen
```

```
2
```

Establir demanda

demanda

Què passa si establim demanda a un punt que no és terminal?

demanda

C1

250

```
1 Error: C1 no és un terminal
```

```
2
```