

Matemàtica computacional i analítica de dades
Algorítmia i combinatòria en grafs ...
Curs 2020–21

Pràctica avaluable

Volem fer un programa que ens resolgui (quan es pugui) el **Problema de les gerres**:

https://en.wikipedia.org/wiki/Water_pouring_puzzle

El problema consisteix en obtenir la quantitat desitjada d'aigua a una de les gerres a partir d'una configuració inicial i seguint les normes següents:

- Sabem la capacitat de cada gerra.
- Podem passar aigua d'una gerra a una altra fins que la primera quedi buida o la segona plena (no ens podem aturar abans).

Per exemple, considerem tres gerres (**gerra0**, **gerra1**, **gerra2**), de les que la **gerra0** és de 8 litres i està plena, la **gerra1** és de 5 litres i està buida i la **gerra3** és de 3 litres i està buida.



Figura 1: Situació inicial, amb la gerra de 8 litres plena i les altres dues buides (Imatge de la wikipedia).

Suposem que volem aconseguir que a una gerra hi hagi 4 litres. Llavors podem fer:

- La situació inicial és $|8|0|0|$.
- Passem aigua de la **gerra0** a la **gerra1**, quedant $|3|5|0|$.
- Passem aigua de la **gerra1** a la **gerra2**, quedant $|3|2|3|$.
- Passem aigua de la **gerra2** a la **gerra0**, quedant $|6|2|0|$.
- Passem aigua de la **gerra1** a la **gerra2**, quedant $|6|0|2|$.
- Passem aigua de la **gerra0** a la **gerra1**, quedant $|1|5|2|$.
- Passem aigua de la **gerra1** a la **gerra2**, quedant $|1|4|3|$, i ja tenim un 4, que és el que volíem.

Podeu trobar aquest joc a (versió Android)

https://play.google.com/store/apps/details?id=com.deadlinefighter.waterjug&hl=en_US&gl=US
o una variant a la web:

<https://www.transum.org/Software/Investigations/jugs.asp>
on, només hi ha dues gerres^{1 2}.

¹El problema el podem pensar amb 3 gerres afegint una gerra nova al principi amb una capacitat i quantitat de líquid que sigui igual o superior a la suma de les altres dues, i deixar les altres dues buides. Per exemple, una situació inicial de $|100|0|0|$ on la capacitat de la **gerra0** és 100, la de la **gerra1** és 7 i la de la **gerra2** és 5, donaria una solució al problema inicial que ens proposa aquesta web. Hauríem d'interpretar que fer transvasaments que afectin a la **gerra0** voldria dir omplir o buidar la gerra corresponent

²Aquesta versió apareix a la pel·lícula *Die Hard: With a Vengeance* (John McTiernan, 1995) on, entre els minuts 59 i 62 de la pel·lícula, John (Bruce Willis) i Zeus (Samuel Jackson) tenen dues gerres de 5 i 3 galons respectivament i han d'aconseguir tenir exactament 4 galons a una de les gerres.

El graf

Per tal de fixar el llenguatge, direm que:

- Una *configuració* és una tripleta $|A|B|C|$, on A , B i C son enters positius que representen els litres d'aigua a cada gerra.
- Per passar d'una configuració a una altra, fem transvasament de la gerra j a la gerra k que notarem com (j, k) .

Cada configuració és un vèrtex del graf, mentre que hi ha una aresta entre dos vèrtexs si hi ha alguna parella de gerres j, k tal que el transvasament que (j, k) ens passa d'una configuració a l'altra.

Per exemple, el graf corresponent a l'exemple que comença amb tres gerres amb capacitats 8, 5 i 3 respectivament, amb la primera plena i les altres buides és:

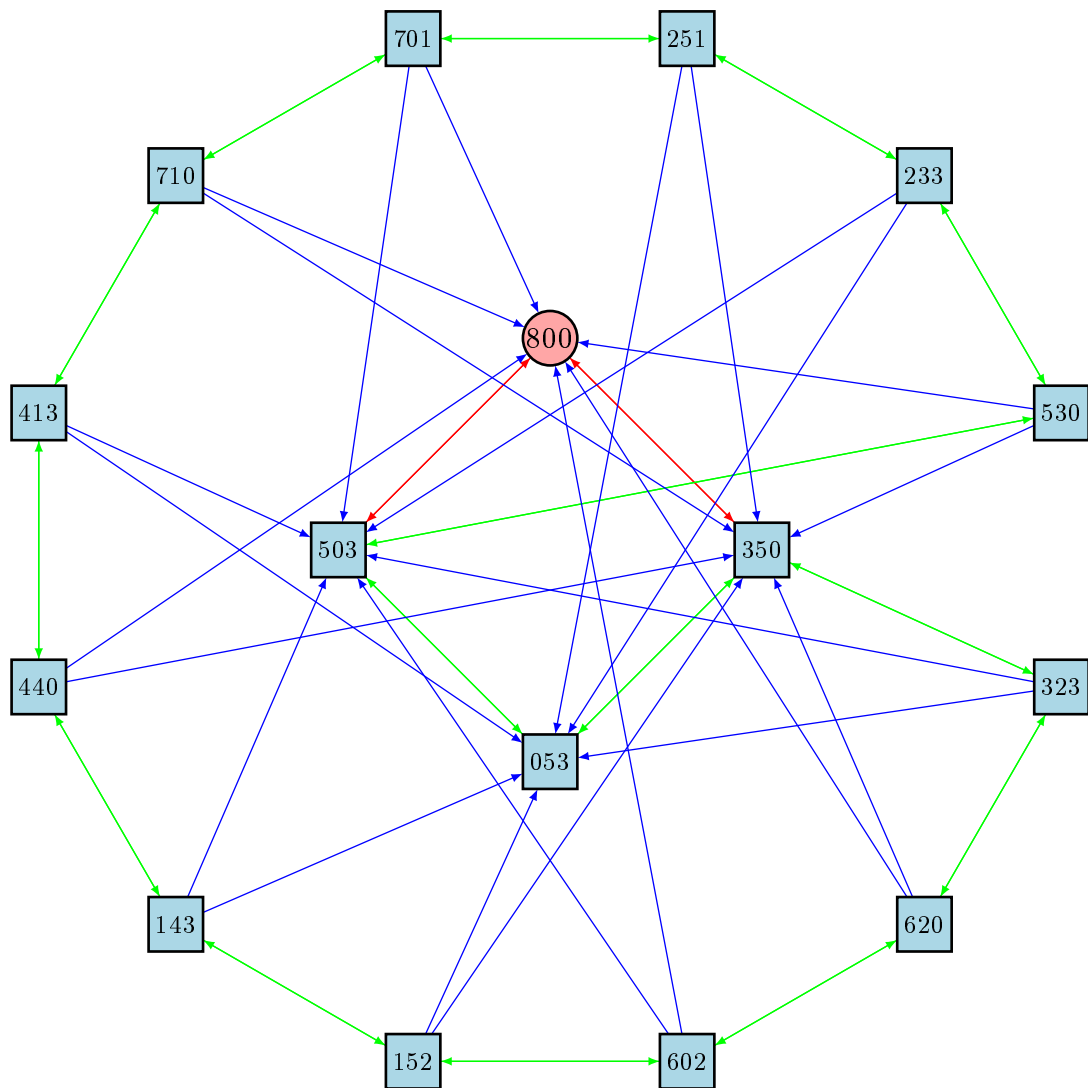


Figura 2: Graf amb totes les configuracions de 3 gerres amb capacitats 8, 5 i 3 litres. Els 3 dígit representen la quantitat de líquid a cada gerra en l'ordre que estan.

En general, no volem descriure tot el graf, sinó que el que volem és saber quines configuracions són possibles i com arribar-hi des de la situació inicial.

Codi inicial

El codi que ve a continuació mostra totes les configuracions possibles del cas anterior, però de forma massa particular, amb limitacions del màxim de litres i configuracions, i poc eficient. A aquesta pràctica modificarem aquest codi per tal d'obtenir la solució per altres casos, sense limitacions fixades pel codi (tindrem les limitacions dels tipus de dades escollits i de l'ordinador), i de forma més eficient.

```
1 #include<stdio.h>
2
3 #define NGERRES 3
4 #define MAXLITRES 10
5 #define MAXCONF 20
6
7 typedef struct{
8     unsigned capacitat,contingut;
9 }gerra;
10
11 typedef struct{
12     unsigned continguts[NGERRES];
13 }configuracio;
14
15 int main(void)
16 {
17     unsigned i,j,k,l;
18     char estat_config[MAXLITRES+1][MAXLITRES+1][MAXLITRES+1];
19     gerra gerres[NGERRES]={8,8},{5,0},{3,0};
20     configuracio configuracions[MAXCONF+1];
21     printf("Gerres amb capacitats |");
22     for(l=0;l<NGERRES;l++) printf("%u|",gerres[l].capacitat);
23     printf("\n");
24     for(i=0;i<gerres[0].capacitat+1;i++){
25         for(j=0;j<gerres[1].capacitat+1;j++){
26             for(k=0;k<gerres[2].capacitat+1;k++){
27                 estat_config[i][j][k]=0;
28             }
29         }
30     }
31     for(l=0;l<NGERRES;l++) configuracions[0].continguts[l]=gerres[l].
contingut;
32     printf("Configuració inicial |");
33     for(l=0;l<NGERRES;l++) printf("%u|",configuracions[0].continguts[l]);
34     printf("\n");
35     estat_config[configuracions[0].continguts[0]][configuracions[0].
continguts[1]][configuracions[0].continguts[2]]=1;
36     int num_conf=1;
37     for(i=0;i<num_conf && num_conf<MAXCONF;i++){
38         for(j=0;j<NGERRES && num_conf<MAXCONF;j++){
39             for(k=0;k<NGERRES && num_conf<MAXCONF;k++){
40                 if(j==k) continue;
41                 if(configuracions[i].continguts[j]==0) continue;
42                 if(configuracions[i].continguts[k]==gerres[k].capacitat)
continue;
43                 for(l=0;l<NGERRES;l++) configuracions[num_conf].continguts[
l]=configuracions[i].continguts[l];
44                 configuracions[num_conf].continguts[k]=configuracions[
num_conf].continguts[k]+configuracions[num_conf].continguts[j];
45                 configuracions[num_conf].continguts[j]=0;
```

```

46         if (configuracions[num_conf].continguts[k]>gerres[k].
capacitat){
47             configuracions[num_conf].continguts[j]=configuracions[
num_conf].continguts[k]-gerres[k].capacitat;
48             configuracions[num_conf].continguts[k]=gerres[k].
capacitat;
49         }
50         if (estat_config[configuracions[num_conf].continguts[0]][
configuracions[num_conf].continguts[1]][configuracions[num_conf].
continguts[2]]==1) continue;
51         estat_config[configuracions[num_conf].continguts[0]][
configuracions[num_conf].continguts[1]][configuracions[num_conf].
continguts[2]]=1;
52         printf("De configuració %u a %u: |",i,num_conf);
53         for (l=0;l<NGERRES;l++) printf("%u|",configuracions[i].
continguts[l]);
54         printf("-(%u,%u)-|",j,k);
55         for (l=0;l<NGERRES;l++) printf("%u|",configuracions[num_conf
].continguts[l]);
56         printf("\n");
57         num_conf++;
58     }
59 }
60 }
61
62 if (num_conf==MAXCONF){
63     printf("Hem arribat al màxim de configuracions i pot ser no s'han
fet totes\n");
64     return 1;
65 }
66 printf("S'han trobat %u configuracions possibles\n",num_conf);
67 return 0;
68 }

```

Comproveu el seu funcionament compilant-lo i executant-lo. Hauria de mostrar la sortida següent:

```

Gerres amb capacitats |8|5|3|
Configuració inicial |8|0|0|
De configuració 0 a 1: |8|0|0|-(0,1)-|3|5|0|
De configuració 0 a 2: |8|0|0|-(0,2)-|5|0|3|
De configuració 1 a 3: |3|5|0|-(0,2)-|0|5|3|
De configuració 1 a 4: |3|5|0|-(1,2)-|3|2|3|
De configuració 2 a 5: |5|0|3|-(2,1)-|5|3|0|
De configuració 4 a 6: |3|2|3|-(2,0)-|6|2|0|
De configuració 5 a 7: |5|3|0|-(0,2)-|2|3|3|
De configuració 6 a 8: |6|2|0|-(1,2)-|6|0|2|
De configuració 7 a 9: |2|3|3|-(2,1)-|2|5|1|
De configuració 8 a 10: |6|0|2|-(0,1)-|1|5|2|
De configuració 9 a 11: |2|5|1|-(1,0)-|7|0|1|
De configuració 10 a 12: |1|5|2|-(1,2)-|1|4|3|
De configuració 11 a 13: |7|0|1|-(2,1)-|7|1|0|
De configuració 12 a 14: |1|4|3|-(2,0)-|4|4|0|
De configuració 13 a 15: |7|1|0|-(0,2)-|4|1|3|
S'han trobat 16 configuracions possibles

```

Si busquem la primera configuració que té un 4, veiem que és la 12, i si mirem endarrere, per arribar a la 12, hem fet:

```

|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|-(2,0)-|6|2|0|-(1,2)-|6|0|2|-(0,1)
-|1|5|2|-(1,2)-|1|4|3|.

```

Exercicis

Afegiu com a les dues primeres línies de cada codi i en format comentari els vostres Nom, Cognom i NIU.

Exercici 1: Guardeu el codi inicial amb el nom **Ex1.c**. Modifiqueu el codi per a que el programa per a que s'entrin 6 paràmetres, corresponents a la capacitat i contingut de cada gerra. Per exemple, si executem:

```
./Ex1 8 8 5 0 3 0
```

hauria de retornar el mateix que hem vist abans. Hauríeu d'evitar que s'executés el programa si s'entren paràmetres incoherents: algun contingut superior a la capacitat, o bé alguna capacitat superior al màxim de litres que admet el programa.

Exercici 2: Guardeu el codi anterior amb el nom **Ex2.c**. Programeu una funció nova anomenada **transvasament** que, cridada com:

```
1 configuracions[num_conf]=transvasament(j,k,configuracions[i],gerres);
```

substitueixi les línies 43–49 del codi original.

Exercici 3: Guardeu el codi anterior amb el nom **Ex3.c**. Per a cada configuració, volem mostrar, de forma ordenada, els transvasaments de continguts que s'ha de fer per arribar-hi des de la situació inicial. Per a fer això, afegim camps a l'estructura configuració: dos **unsigned** que ens diuen de quina gerra a quina gerra passem el líquid, i un altre dient quin és el número de la configuració anterior:

```
1 typedef struct{
2     unsigned continguts[NGERRES];
3     unsigned origen, desti;
4     unsigned anterior;
5 }configuracio;
```

Volem mostrar el camí mitjançant una funció que ens mostri com s'ha arribat a aquesta a partir de la configuració inicial. La funció pot tenir el prototipus:

```
1 void mostracami(unsigned, configuracio*)
```

i mostrarà con s'arriba a la configuració donada per l'enter sense signe.

Exercici 4: Guardeu el codi anterior amb el nom **Ex4.c**. Volem generalitzar el programa per a que contempli el cas de n gerres enlloc de 3, i que no hi hagi un màxim de litres per gerra prefixat. Això vol dir que el primer paràmetre que passarem al programa ara és el nombre de gerres. Per exemple, cridarem:

```
./Ex4 3 8 8 5 0 3 0
```

Modifiqueu el codi per a que funcioni per a n gerres en general i sense màxim de litres. Heu de tenir en compte que caldrà modificar tot el codi que utilitzi **NGERRES** i **MAXLITRES**, així com afegir un paràmetre amb el nombre de gerres a les funcions que hagueu programat anteriorment. En particular cal modificar l'estructura:

```
1 typedef struct config{
2     unsigned *continguts;
3 }configuracio;
```

i reservar tantes posicions com facin falta cada cop que es troba una configuració nova. També cal tenir en compte que el vector amb 3 índexs `estat_config` està programat exactament per a 3 gerres, i necessitarà una modificació més profunda que s'adapti a qualsevol nombre de gerres (**Indicació:** una opció és passar de 3 índexs a un de sol que prengui un valor entre 0 i el nombre de configuracions possibles).

Exercici 5: Guardeu el codi anterior amb el nom `Ex5.c`. Volem eliminar que hi hagi un màxim de configuracions. Per a fer això, modifiquem la manera com guardem les configuracions. Ara s'està fent amb un vector `configuracions[MAXCONF+1]`, i, per tal de que quedi més òptim, haurem de guardar les configuracions en una llista enllaçada. A més, per a poder trobar totes les configuracions haurem de seguir les configuracions amb una cua. Per a això, utilitzeu les estructures:

```
1 typedef struct elementcua{
2     configuracio *config;
3     struct elementcua *seguent;
4 }elementcua;
5
6 typedef struct{
7     elementcua *primera;
8     elementcua *ultima;
9 }cua;
```

I també funcions `encua` i `desencua` que permetin afegir un element al final de la cua i fer que la cua comenci després de l'element que acabem de desplegar.

Donat que ara les configuracions no tindran un número assignat, haureu de modificar part del codi i de l'estructura `configuracio`, quedant:

```
1 typedef struct config{
2     unsigned *continguts;
3     unsigned origen, desti;
4     struct config *anterior;
5 }configuracio;
```

Aquests canvis també afectaran a altres funcions que hagueu programat (com per exemple `mostracami` i `transvasament`).

Exercici 6: Guardeu el codi anterior amb el nom `Ex6.c`. Finalment, volem que tant sols ens mostri el camí que porta a una configuració que contingui el nombre de litres desitjat, i que, en cas de que no es pugui, mostri un missatge d'error. Ens limitarem a entrar un o més paràmetres al final de tot i que mostri el camí més curt que ens porta a una configuració amb gerres amb el nombre de litres d'aquests paràmetres que hem entrat. Per exemple:

```
./Ex6 3 8 8 5 0 3 0 4
```

només ens mostra el primer camí que troba que té una gerra amb 4 litres. En canvi,

```
./Ex6 4 10 10 10 10 0 9 0 4 1
```

considerarà 4 gerres de 10, 10, 10 i 9 litres respectivament, amb les 2 primeres plenes, les 2 últimes buides i només ens mostrarà el primer camí que tingui una gerra amb 4 litres i una altra amb 1 litre.

Instruccions finals

A continuació teniu diverses execucions dels programes que hem fet i la corresponent sortida (el \$ que hi ha a l'inici de les línies amb la instrucció és el símbol del sistema). És

important que els vostres codis retornin les respostes en el mateix format i, a la correcció, comprovarem el funcionament dels vostres codis amb altres configuracions.

```
$ ./Ex1 8 8 5 0 3 0
Gerres amb capacitats |8|5|3|
Configuració inicial |8|0|0|
De configuració 0 a 1: |8|0|0|-(0,1)-|3|5|0|
De configuració 0 a 2: |8|0|0|-(0,2)-|5|0|3|
De configuració 1 a 3: |3|5|0|-(0,2)-|0|5|3|
De configuració 1 a 4: |3|5|0|-(1,2)-|3|2|3|
De configuració 2 a 5: |5|0|3|-(2,1)-|5|3|0|
De configuració 4 a 6: |3|2|3|-(2,0)-|6|2|0|
De configuració 5 a 7: |5|3|0|-(0,2)-|2|3|3|
De configuració 6 a 8: |6|2|0|-(1,2)-|6|0|2|
De configuració 7 a 9: |2|3|3|-(2,1)-|2|5|1|
De configuració 8 a 10: |6|0|2|-(0,1)-|1|5|2|
De configuració 9 a 11: |2|5|1|-(1,0)-|7|0|1|
De configuració 10 a 12: |1|5|2|-(1,2)-|1|4|3|
De configuració 11 a 13: |7|0|1|-(2,1)-|7|1|0|
De configuració 12 a 14: |1|4|3|-(2,0)-|4|4|0|
De configuració 13 a 15: |7|1|0|-(0,2)-|4|1|3|
S'han trobat 16 configuracions possibles

$ ./Ex1 8 8 5 7 3 0
Configuració no possible a la gerra 1.

$ ./Ex1 12 8 5 0 3 0
Configuració no possible a la gerra 0.

$ ./Ex3 8 8 5 0 3 0
Gerres amb capacitats |8|5|3|
Configuració inicial |8|0|0|
|8|0|0|-(0,1)-|3|5|0| | | | | | | | | | | | |
|8|0|0|-(0,2)-|5|0|3|
|8|0|0|-(0,1)-|3|5|0|-(0,2)-|0|5|3|
|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|
|8|0|0|-(0,2)-|5|0|3|-(2,1)-|5|3|0|
|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|-(2,0)-|6|2|0|
|8|0|0|-(0,2)-|5|0|3|-(2,1)-|5|3|0|-(0,2)-|2|3|3|
|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|-(2,0)-|6|2|0|-(1,2)-|6|0|2|
|8|0|0|-(0,2)-|5|0|3|-(2,1)-|5|3|0|-(0,2)-|2|3|3|-(2,1)-|2|5|1|
|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|-(2,0)-|6|2|0|-(1,2)-|6|0|2|-(0,1)
-|1|5|2|
|8|0|0|-(0,2)-|5|0|3|-(2,1)-|5|3|0|-(0,2)-|2|3|3|-(2,1)-|2|5|1|-(1,0)
-|7|0|1|
|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|-(2,0)-|6|2|0|-(1,2)-|6|0|2|-(0,1)
-|1|5|2|-(1,2)-|1|4|3|
|8|0|0|-(0,2)-|5|0|3|-(2,1)-|5|3|0|-(0,2)-|2|3|3|-(2,1)-|2|5|1|-(1,0)
-|7|0|1|-(2,1)-|7|1|0|
|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|-(2,0)-|6|2|0|-(1,2)-|6|0|2|-(0,1)
-|1|5|2|-(1,2)-|1|4|3|-(2,0)-|4|4|0|
|8|0|0|-(0,2)-|5|0|3|-(2,1)-|5|3|0|-(0,2)-|2|3|3|-(2,1)-|2|5|1|-(1,0)
-|7|0|1|-(2,1)-|7|1|0|-(0,2)-|4|1|3|
S'han trobat 16 configuracions possibles

$ ./Ex4 4 8 8 5 0 3 0 2 0
Gerres amb capacitats |8|5|3|2|
Configuració inicial |8|0|0|0|
|8|0|0|0|-(0,1)-|3|5|0|0| | | | | |
|8|0|0|0|-(0,2)-|5|0|3|0|
|8|0|0|0|-(0,3)-|6|0|0|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,2)-|0|5|3|0|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,3)-|1|5|0|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(1,2)-|3|2|3|0|
```

```

|8|0|0|0|-(0,1)-|3|5|0|0|-(1,3)-|3|3|0|2| | | | | |
|8|0|0|0|-(0,2)-|5|0|3|0|-(0,3)-|3|0|3|2|
|8|0|0|0|-(0,2)-|5|0|3|0|-(2,1)-|5|3|0|0|
|8|0|0|0|-(0,2)-|5|0|3|0|-(2,3)-|5|0|1|2|
|8|0|0|0|-(0,3)-|6|0|0|2|-(3,1)-|6|2|0|0|
|8|0|0|0|-(0,3)-|6|0|0|2|-(3,2)-|6|0|2|0|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,2)-|0|5|3|0|-(1,3)-|0|3|3|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,2)-|0|5|3|0|-(2,3)-|0|5|1|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,3)-|1|5|0|2|-(1,2)-|1|2|3|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,3)-|1|5|0|2|-(3,2)-|1|5|2|0|
|8|0|0|0|-(0,1)-|3|5|0|0|-(1,2)-|3|2|3|0|-(2,3)-|3|2|1|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(1,3)-|3|3|0|2|-(3,2)-|3|3|2|0|
|8|0|0|0|-(0,2)-|5|0|3|0|-(2,1)-|5|3|0|0|-(0,2)-|2|3|3|0|
Hem arribat al màxim de configuracions i pot ser no s'han fet totes

```

```

$ ./Ex5 4 8 8 5 0 3 0 2 0
Gerres amb capacitats |8|5|3|2|
Configuració inicial |8|0|0|0|
|8|0|0|0|-(0,1)-|3|5|0|0| | | | | |
|8|0|0|0|-(0,2)-|5|0|3|0|
|8|0|0|0|-(0,3)-|6|0|0|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,2)-|0|5|3|0|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,3)-|1|5|0|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(1,2)-|3|2|3|0|
|8|0|0|0|-(0,1)-|3|5|0|0|-(1,3)-|3|3|0|2|
|8|0|0|0|-(0,2)-|5|0|3|0|-(0,3)-|3|0|3|2|
...
|8|0|0|0|-(0,2)-|5|0|3|0|-(2,1)-|5|3|0|0|-(0,2)-|2|3|3|0|-(1,3)
    -|2|1|3|2|-(3,0)-|4|1|3|0|-(2,3)-|4|1|1|2|
|8|0|0|0|-(0,1)-|3|5|0|0|-(0,3)-|1|5|0|2|-(1,2)-|1|2|3|2|-(3,1)
    -|1|4|3|0|-(2,3)-|1|4|1|2|-(3,1)-|1|5|1|1|-(1,0)-|6|0|1|1|
|8|0|0|0|-(0,1)-|3|5|0|0|-(1,3)-|3|3|0|2|-(3,2)-|3|3|2|0|-(1,3)
    -|3|1|2|2|-(3,2)-|3|1|3|1|-(2,0)-|6|1|0|1|-(3,2)-|6|1|1|0|
S'han trobat 60 configuracions possibles

```

```

$ ./Ex6 3 8 8 5 0 3 0 4
Gerres amb capacitats |8|5|3|
Configuració inicial |8|0|0|
|8|0|0|-(0,1)-|3|5|0|-(1,2)-|3|2|3|-(2,0)-|6|2|0|-(1,2)-|6|0|2|-(0,1)
    -|1|5|2|-(1,2)-|1|4|3|

```

```

$ ./Ex6 3 8 8 5 0 3 0 9
Gerres amb capacitats |8|5|3|
Configuració inicial |8|0|0|
S'han trobat 16 configuracions, però cap té |9| litres.

```

```

$ ./Ex6 4 10 10 10 10 10 0 9 0 1 4
Gerres amb capacitats |10|10|10|9|
Configuració inicial |10|10|0|0|
|10|10|0|0|-(0,3)-|1|10|0|9|-(3,2)-|1|10|9|0|-(0,3)-|0|10|9|1|-(1,3)
    -|0|2|9|9|-(3,0)-|9|2|9|0|-(0,2)-|8|2|10|0|-(1,3)-|8|0|10|2|-(2,3)
    -|8|0|3|9|-(3,0)-|10|0|3|7|-(3,1)-|10|7|3|0|-(0,3)-|1|7|3|9|-(0,2)
    -|0|7|4|9|-(3,0)-|9|7|4|0|-(0,1)-|6|10|4|0|-(1,3)-|6|1|4|9|

```

Feu el lliurament dels fitxers de codi a través de l'apartat corresponent del Campus Virtual de l'assignatura. Recordeu que al principi de cada fitxer hi ha d'haver el vostre nom i NIU.