Universitat de Girona
Escola Politècnica Superior

Exercici 5 de laboratori

Estructures de Dades i Algorítmica

Curs 2025/26

Universitat de Girona
Departament d'Informàtica, Matemàtica Aplicada i Estadística

Jerónimo Hernández González
jeronimo.hernandez@udg.edu

(PDF creat el 24 d'octubre de 2025)

Índex

1 Presentació

L'aeroport de Girona ens ha demanat un programa per gestionar els vols i el seu ús de les portes d'embarcament a la terminal. S'ha de gestionar una sèrie de vols, $v = \{1, 2, \ldots, n\}$, de cadascun dels quals sabem:

- la seva destinació (regional o internacional), i
- l'hora de sortida.

També sabem que l'aeroport:

- disposa de 2 portes amb control de passaports (aptes per a vols internacionals) i 3 més per a vols regionals.
- reserva la porta durant 60 minuts abans de la sortida de cada vol per al seu servei.

Hem de buscar una assignació de vols a portes (i horari) tenint en compte que:

- els vols internacionals només es podran embarcar des d'una porta amb control de passaport. Els vols regionals poden operar-se des de qualsevol porta.
- s'han d'assignar tots els vols a una porta d'embarcament compatible (que estigui disponible i pugui operar el tipus de vol) fent servir el **mínim nombre de portes possible**.

Ens demanen un programa que faci aquesta assignació. Els hem comunicat que, donat la complexitat del problema, la solució òptima pot tenir un alt cost computacional. Ens demanen la **possibilitat de triar** una solució qualsevol vàlida, o buscar l'opció òptima.

1.1 Característiques del programa

Així, heu de fer un programa que:

- Rebi per paràmetre des de la terminal els valors que configuren el problema:

`-r <int>` : nombre de portes per a vols exclusivament regionals. Valor per defecte, 3.

`-i <int>` : nombre de portes amb control de passaport (necessari pels vols internacionals). Valor per defecte, 2.

`fitxer_dades` : el nom (i ruta) del fitxer de dades que es vol fer servir

- Llegeixi els vols del fitxer indicat. Vegeu-ne un exemple a la Secció 3.2.

  Podeu fer els preprocessaments de les dades que necessiteu.

- Implementi, seguint les variants de l'esquema de *backtracking*, un algoritme que torni una solució vàlida per al problema i un altre que torni la solució òptima.

  L'usuari podrà triar quin algoritme s'executa des de la terminal:

"" : per defecte (no caldrà passar cap opció), s'executarà l'algoritme que torna **una solució vàlida** (la primera).

`-m` : s'executarà l'algoritme que torna la **millor solució**.

- Mostri per pantalla la **distribució de vols** en portes proposta. Vegeu-ne un exemple a la Secció 3.3.

  També mostrareu el **temps de còmput** necessari per trobar la solució.

2 Lliurament

El dia de l'entrega haureu de tenir el vostre programa a `bas.udg.edu`, preparat per compilar i executar. Això inclou:

1. el codi font (només fitxers `.cpp` i `.h`, res de fitxers objecte ni executables)
2. els fitxers del joc de proves (veure Sec. 3),
3. el fitxer `llegeix.me` on expliqueu quin objectiu té cada fitxer del joc de proves (i qualsevol altre comentari que vulgueu fer sobre el vostre codi).

IMPORTANT: Cal que seguiu les instruccions sobre com lliurar les activitats de laboratori que teniu a Moodle. Assegureu-vos que ho feu com us demanem, sobretot les pre- i postcondicions tant a les classes com al `main`.

3 Joc de proves i sortida

Cal que acompanyeu el vostre codi amb un joc de proves. En aquest cas, el joc de proves es redueix a diferents fitxers de dades. Com sempre, us donem un joc de proves bàsic que **heu de complementar**.

El joc de proves que compartim a Moodle consta de 2 fitxers de dades. També hi trobareu les corresponents sortides possible. Vigileu perquè, també en aquest cas, segons com tracta el problema l'algoritme de *backtracking* que implementeu, la sortida que obteniu pot ser diferent a la que us donem. En el cas de l'algoritme que retorna una solució, la sortida pot ser molt diferent. En el cas de l'algoritme que retorna la millor solució, aquesta també pot ser diferent a la que us donem però ha de ser equivalent d'acord amb el criteri d'optimització (mateix nombre de portes).

A continuació reproduim el funcionament bàsic del programa, el contingut de `volsCurt.txt` i la sortida de la nostra implementació dels algoritmes de *backtracking* corresponent.

3.1 Funcionament bàsic

```
$ ./e5
Falten arguments ("e5 --help" per ajuda)
```

```
$ ./e5 --help
Ús: ./e5 [-h] | [-m] [-r <int>] [-i <int>] fitxer

opcio pot ser:
  -h, --help      mostra aquest missatge d'ajuda i surt

  -m              cerca la solució que minimitza el nombre de portes
  -r <int>        indica el nombre de portes de tipus regional disponibles
  -i <int>        indica el nombre de portes internacionals disponibles

fitxer           fitxer de text amb tots els vols i altres dades requerides
```

```
$ ./e5 -r -i 3
Error: El valor associat a l'opció '-r' és incorrecte.
```

```
$ ./e5 -r 2
Error: Falta el nom del fitxer.
```

```
$ ./e5 -r 2 noexisteix.txt
Error: El fitxer [noexisteix.txt] no es pot obrir. Repassa el nom i permisos.
```

3.2 Fitxer de proves: `volsCurt.txt`

```
id tipus hora
1 i 07:30
2 r 07:45
3 r 08:30
4 r 09:00
5 i 09:30
6 i 10:00
7 r 10:15
8 i 11:00
9 i 11:30
10 r 11:30
11 i 12:45
12 r 13:30
13 i 13:45
```

Podeu reaprofitar el codi de `eines.h` per llegir aquestes dades (el separador és el tabulador, '\t').

3.3 Execució amb: `volsCurt.txt`

```
$ ./e5 volsCurt.txt
==> 13 vols llegits.

************************
* Porta 1 (REG): 4 vols *
*----------------------*
* Vol:  1 [07:30] REG   *
* Vol:  6 [10:00] REG   *
* Vol:  8 [11:15] REG   *
* Vol: 11 [12:45] REG   *
************************

************************
* Porta 2 (REG): 3 vols *
*----------------------*
* Vol:  2 [07:45] REG   *
* Vol:  7 [10:15] REG   *
* Vol: 12 [13:30] REG   *
************************

************************
* Porta 3 (INT): 4 vols *
*----------------------*
* Vol:  3 [08:45] INT   *
* Vol:  5 [09:45] INT   *
* Vol:  9 [11:30] INT   *
* Vol: 13 [13:45] INT   *
************************

************************
* Porta 4 (INT): 2 vols *
*----------------------*
* Vol:  4 [09:00] INT   *
* Vol: 10 [11:30] INT   *
************************

Num. portes: 4 (INT.: 2)
Temps: 0.000292767 segons
```

```
$ ./e5 -m volsCurt.txt
==> 13 vols llegits.

************************
* Porta 1 (INT): 6 vols *
*----------------------*
* Vol:  1 [07:30] REG   *
* Vol:  3 [08:45] INT   *
* Vol:  5 [09:45] INT   *
* Vol:  9 [11:30] INT   *
* Vol: 11 [12:45] REG   *
* Vol: 13 [13:45] INT   *
************************

************************
* Porta 2 (INT): 5 vols *
*----------------------*
* Vol:  2 [07:45] REG   *
* Vol:  4 [09:00] INT   *
* Vol:  6 [10:00] REG   *
* Vol: 10 [11:30] INT   *
* Vol: 12 [13:30] REG   *
************************

************************
* Porta 3 (REG): 2 vols *
*----------------------*
* Vol:  7 [10:15] REG   *
* Vol:  8 [11:15] REG   *
************************

Num. portes: 3 (INT.: 2)
Temps: 0.00251557 segons
```

3.4 Execució amb: `volsLlarg.txt`

```
$ ./e5 volsLlarg.txt
==> 40 vols llegits.
************************
* Porta 1 (INT):13 vols *
*----------------------*
* Vol:  1 [07:30] INT   *
* Vol:  4 [08:30] REG   *
* Vol:  7 [09:30] INT   *
* Vol: 11 [11:00] INT   *
* Vol: 14 [12:30] INT   *
* Vol: 17 [13:30] INT   *
* Vol: 20 [15:30] REG   *
* Vol: 23 [16:30] REG   *
* Vol: 26 [17:30] INT   *
* Vol: 29 [18:30] REG   *
* Vol: 32 [19:30] REG   *
* Vol: 36 [20:30] INT   *
* Vol: 39 [21:30] INT   *
************************
************************
* Porta 2 (REG):11 vols *
*----------------------*
* Vol:  2 [07:45] REG   *
* Vol:  6 [09:00] REG   *
* Vol:  9 [10:00] REG   *
* Vol: 13 [11:30] REG   *
* Vol: 15 [12:45] REG   *
* Vol: 19 [14:15] REG   *
* Vol: 22 [16:00] REG   *
* Vol: 24 [17:00] REG   *
* Vol: 30 [18:30] REG   *
* Vol: 34 [20:00] REG   *
* Vol: 37 [21:00] REG   *
************************
************************
* Porta 3 (REG): 6 vols *
*----------------------*
* Vol:  3 [07:30] REG   *
* Vol: 10 [10:15] REG   *
* Vol: 16 [12:45] REG   *
* Vol: 25 [17:00] REG   *
* Vol: 35 [20:15] REG   *
* Vol: 33 [21:15] REG   *
************************
************************
* Porta 4 (INT): 9 vols *
*----------------------*
* Vol:  5 [08:30] INT   *
* Vol:  8 [10:00] INT   *
* Vol: 12 [11:30] INT   *
* Vol: 18 [13:45] INT   *
* Vol: 21 [15:30] INT   *
* Vol: 27 [17:30] INT   *
* Vol: 31 [19:00] INT   *
* Vol: 33 [20:00] INT   *
* Vol: 40 [21:45] REG   *
************************
************************
* Porta 5 (REG): 1 vols *
*----------------------*
* Vol: 28 [17:45] REG   *
************************

Num. portes: 5 (INT.: 2)
Temps: 0.000434927 segons
```

```
$ ./e5 -m volsLlarg.txt
==> 40 vols llegits.

...cancel·lat després de 4 hores
executant-se sense resposta...
```