

ANÀLISI I DISSENY D'ALGORISMES

Mètode voraç

Pràctica 7 de laboratori

Lliurament: Fins a diumenge 23 d'abril, 23.55h. A través de Moodle

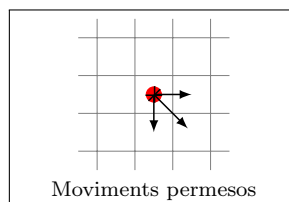
El problema del laberint II

[L'enunciat del problema és idèntic al de la pràctica anterior; la diferència està en la manera d'abordar-lo i en la solució obtinguda, que en aquest cas pot ser que no siga l'òptima o que no trobe camí d'eixida encara que n'hi haja.]

Es disposa de una quadrícula $n \times m$ de valors $\{0, 1\}$ que representa un laberint. Un valor 0 en una casella qualsevol de la quadrícula indica una posició inaccessible; en canvi, amb el valor 1 s'indiquen les caselles accessibles. Per exemple:

$\xrightarrow{(0,0)}$	1	1	0	0	1
	1	1	1	1	1
	0	1	1	0	0
	1	1	0	1	1
	1	1	1	0	0
	0	0	0	1	1
					$\xrightarrow{(5,4)}$

Un laberint 6 x 5



Es demana aplicar el mètode voraç¹ per a tractar d'obtenir la longitud² del camí més curt que partint de l'origen $(0, 0)$ condueix a la destinació $(n - 1, m - 1)$ assumint només tres possibles moviments des d'una casella qualsevol (i, j) :

1. cap a la dreta: $(i, j + 1)$,
2. cap avall: $(i + 1, j)$,
3. cap avall i cap a la dreta (diagonal): $(i + 1, j + 1)$.

Evidentment, no són vàlids tampoc els moviments que porten a l'exterior del laberint ni els que condueixen a caselles inaccessibles.

■ Noms, en el codi font, d'algunes funcions importants.

La funció que, seguint una estratègia voraç, decideix les caselles que componen el camí s'ha de dir `maze_greedy`.

Si aquesta funció no es troba en el codi (o hi és, però amb un altre nom) llavors es considerarà que no ha estat desenvolupada, amb la minva consegüent en la qualificació d'aquesta pràctica.

¹S'ha de tenir en compte que el mètode voraç no garanteix, per a aquest problema, que es trobe la solució òptima; ni tan sols que es trobe un camí de eixida encara que n'hi haja.

²Definim *longitud del camí* com el nombre de caselles que el componen. Un camí amb origen i destinació en la mateixa casella té longitud 1.

Es deixa total llibertat per a afegir els paràmetres que es consideren oportuns, i per a triar el tipus de dades de retorn, les estructures de dades, així com les llibreries, els elements del llenguatge, etc.

■ Nom del programa, opcions i sintaxi de l'ordre.

El programa que heu d'escriure s'ha de dir `maze_greedy`. L'ordre tindrà la sintaxi següent:

```
maze_greedy [-p] -f fitxer_entrada
```

- L'opció `-f`, l'única obligatòria, s'utilitza per a subministrar-hi el nom del fitxer que conté la instància del problema a resoldre. En el cas que no s'hi subministre o s'hi produïska algun tipus d'error en tractar d'obrir-lo (perquè no existeix, perquè no es té permís per a obrir-ho, etc.) s'anunciarà amb un missatge d'error. No és necessari controlar possibles errors en el contingut del fitxer d'entrada, ja que aquest sempre s'ajustarà fidelment al format establert, el qual es descriu a l'apartat "Entrada al programa".
- L'opció `-p` s'utilitzarà per a mostrar el camí proposat.
- Totes dues opcions podran aparèixer en qualsevol ordre.
- En el cas que s'use l'ordre amb una sintaxi diferent a la descrita s'emetrà un missatge d'error que n'advertirà i a continuació es mostrarà la sintaxi correcta. S'han de considerar en aquest cas únicament les opcions inexistents; la duplictat d'opcions pot ser deixada de banda i tractada, per tant, com si l'opció s'haguera escrit només una vegada. No és necessari indicar tots els errors sintàctics que puga contenir l'ordre: hi ha prou amb fer-ho només amb el primer error que es detecte.
- Per a tots els missatges d'error, fins i tot aquell que informa de l'ús apropiat de l'ordre, s'ha d'usar l'eixida estàndard d'error, és a dir, `cerr`.
- Davant qualsevol circumstància d'error en les opcions, el programa haurà d'acabar després d'advertir-ne segons s'acaba de descriure.

■ Eixida del programa i descripció de les opcions:

Si no s'usa l'opció `-p`, el programa mostrarà, en la primera línia de l'eixida, la longitud del millor camí trobat seguint un criteri voraç de decisions. En el cas que aquest criteri no conduïska a l'eixida (o que no s'hi haja camí de eixida) es mostrarà el valor 0.

Si es fa ús de l'opció `-p`, s'afegirà a l'eixida anterior (a partir de la segona línia), el camí recorregut superposat al laberint d'entrada, mitjançant el caràcter `*`. Les altres caselles es deixaran amb el seu valor original. No ha d'haver-hi cap caràcter separador entre tots els valors de cada fila (és a dir, tots els elements d'una mateixa fila es mostraran junts). Al final de cada fila de la matriu ha d'haver-hi un únic salt de línia. (fins aquí, exactament igual que en la pràctica anterior).

No obstant això, s'haurà de mostrar el camí recorregut encara que no s'haja pogut completar el camí fins a l'eixida (vegeu els exemples publicats).

■ Entrada al programa

El laberint $n \times m$ s'hi subministrarà codificat en un fitxer de text el nom del qual es recollirà a través de l'opció `-f`. El seu format i contingut serà:

- Línia 1 del fitxer: valors n i m separats mitjançant un únic espai en blanc.
- Línia 2 (i següents): m valors $\{0,1\}$ que componen la primera fila (i següents) del laberint, separats mitjançant un únic espai en blanc.

Per tant, el fitxer contindrà $n + 1$ línies que acabaran, totes elles, en un salt de línia.

A través de *Moodle* es pot descarregar un arxiu comprimit amb diversos exemples (`?.maze`), juntament amb possibles solucions (`?.maze.sol_greedy`) per al cas que s'use l'opció `-p`.³ És important saber que el fet que el programa que escrigueu obtinga l'eixida correcta per a tots aquests exemples no és garantia que l'obtinga per a uns altres.

■ Format d'eixida. Exemples d'execució.

Entre els exemples d'entrada publicats, està el fitxer `02.maze`, el contingut del qual és el laberint (6×5) utilitzat en la presentació d'aquest problema, que no té solució voraç; i el fitxer `11.maze`, el contingut del qual és un laberint (6×6) que té solució voraç.

A continuació es mostren possibles maneres d'utilitzar en la terminal l'ordre descrita i l'eixida que el programa ha de mostrar (en els exemples, les eixides `cout` i `cerr` estan dirigides a la terminal —seguint el comportament predeterminat—).

És imprescindible cenyir-se al format i text d'eixida mostrats, fins i tot pel que fa als salts de línia o caràcter separador, que en tots els casos és l'espai en blanc. En cap cas s'hi ha d'afegir text o valors addicionals.

ALGUNS EXEMPLES DE SITUACIONS D'ERROR	
<pre>\$maze_greedy -f 02.maze 0 \$maze_greedy -p -f 02.maze 0 *1001 1*111 01*00 110** 11100 00011 \$maze_greedy -f 11.maze 8 \$maze_greedy -f 11.maze -p 8 *01111 *01111 *00001 1*1111 11*111 111***</pre>	<pre>\$maze_greedy -f ERROR: missing filename. Usage: maze_greedy [-p] -f file \$maze_greedy Usage: maze [-p] -f file \$maze_greedy -f -f ERROR: can't open file: -f. Usage: maze_greedy [-p] -f file \$maze_greedy -f 0.problem -p -a -b ERROR: unknown option -a. Usage: maze_greedy [-p] -f file \$maze_greedy -p -f anonexistantfile ERROR: can't open file: anonexistantfile. Usage: maze_greedy [-p] -f file</pre>

³Fixeu-vos que les solucions poden no ser úniques i per tant, cadascuna en particular pot no coincidir amb la publicada; no per això és incorrecta.

■ Normes per al lliurament.

ATENCIÓ: Aquestes normes són de compliment obligatori perquè aquesta pràctica siga avaluada.

És imprescindible cenyir-se al format i al text d'eixida descrits, fins i tot pel que fa als salts de línia o caràcter separador, que en tots els casos és l'espai en blanc. No s'hi ha d'afegir text o valors addicionals. Si aquests requisits no es compleixen, és possible que falle una de les fases de l'avaluació d'aquest treball, que es fa de manera automàtica; en aquest cas, la pràctica no serà avaluada tampoc.

1. El programa s'ha d'escriure com un únic arxiu font amb nom `maze_greedy.cc`.
2. Heu de lliurar únicament els fitxers `maze_greedy.cc` i `makefile` (per a generar l'arxiu executable a través de l'ordre `make` sense afegir-hi res més). **Seguiu estrictament els noms de fitxers i funcions que s'indiquen en aquest enunciat. Només cal lliurar aquests dos arxius (no es lliuraran en cap cas arxius de *test*).**
3. És imprescindible que no presente errors ni de compilació ni d'interpretació (segons corresponga), en els ordinadors del laboratori assignat i en el sistema operatiu *GNU/Linux*.⁴ Es tractarà d'evitar també qualsevol tipus d'avís (*warning*).
4. Tots els fitxers que es lliuren han de contenir el nom de l'autor i el seu DNI (o NIE) en la primera línia (entre comentaris apropiats segons el tipus d'arxiu).
5. Es comprimiran en un arxiu `.tar.gz` el nom del qual serà el DNI de l'alumne, compost de 8 dígits i una lletra (o NIE, compost d'una lletra seguida de 7 dígits i una altra lletra). Per exemple: `12345678A.tar.gz` o `X1234567A.tar.gz`. **Només s'admet aquest format de compressió i només és vàlida aquesta manera d'anomenar l'arxiu.**
6. En l'arxiu comprimit **no hi ha d'haver subcarpetes**; és a dir, en extraure'n els arxius, aquests han de quedar guardats en la mateixa carpeta on es troba l'arxiu que els conté.
7. La pràctica s'ha de pujar a *Moodle* respectant les dates exposades en l'encapçalat d'aquest enunciat.

⁴Si treballem amb el vostre propi ordinador o amb un altre sistema operatiu, assegureu-vos que aquest requisit es compleix (podeu comprovar-ho usant el compilador *online* de <https://godbolt.org>).