

SECONDO PROGETTO ASD 2019/2020

STAR WARS

IL RISVEGLIO DELL'ALGORITMO
(LA GARA DEGLI SGUSCI)

ALLA RICERCA DEI JEDI

In una galassia lontana lontana il Maestro Jedi Obi-Al Montresor e i suoi due padawan stanno esplorando l'universo in cerca di nuovi cavalieri Jedi.



LA FORZA E L'ALGORITMO

Anni di studi da parte dei Jedi hanno portato alla scoperta che una componente fondamentale della Forza è l'**Algoritmo**: per potere diventare dei Jedi è necessario conoscere le vie della Forza e per farlo è *necessario conoscere gli algoritmi*.



IL LATO OSCURO DELLA FORZA

La mancata conoscenza degli algoritmi porta al **Lato Oscuro della Forza**, come dice il Maestro Montresor: «Gli algoritmi inefficienti portano al Lato Oscuro. Gli algoritmi inefficienti conducono all'ira, l'ira all'odio; l'odio conduce alla sofferenza.»



GARA DI SGUSCI

Per mostrare il vostro valore come Jedi, sareste sottoposti a una prova.

Come *Anakin Skywalker*, dovrete partecipare a una **gara di sgusci**. Grazie all'aiuto di una vecchia conoscenza, il pilota di X-Wing Cristian Solo, vi viene fornita una mappa dell'area in cui si svolgerà la gara.

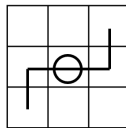
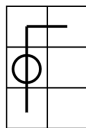
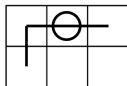


In questa specialità valgono le seguenti regole:

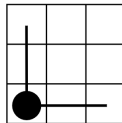
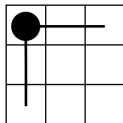
- R1) La gara si svolge su una mappa prestabilita, suddivisa in quadrati, alcuni dei quali possono contenere degli anelli.
- R2) Ogni anello può essere di colore bianco oppure nero.
- R3) L'obiettivo è quello di creare un percorso sulla mappa che connetta più anelli possibili tra loro, formando un **percorso continuo, non auto-intersecante, e preferibilmente chiuso**.

GARA DI SGUSCI - REGOLE

- R4) Se durante il percorso si raggiunge un **anello bianco**, questo deve essere attraversato in maniera rettilinea, curvando nel quadrato immediatamente precedente **e/o** successivo. Ad esempio:

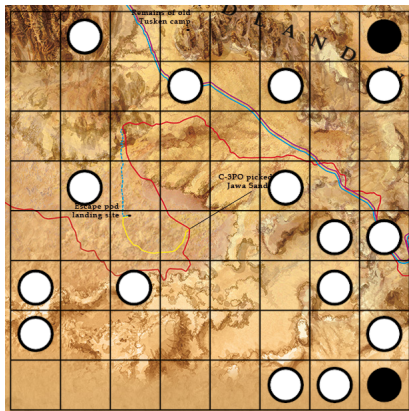


- R5) Se durante il percorso si raggiunge un **anello nero**, al contrario, attraversandolo si deve curvare di novanta gradi, procedendo in modo rettilineo nei quadrati precedente **e** successivo. Ad esempio:



LA GARA HA INIZIO

Per iniziare vi verrà fornita la mappa $N \times M$ dell'area dove si svolgerà la gara, costituita da B anelli neri e W anelli bianchi, di cui vi saranno comunicate le posizioni (r, c) (riga e colonna).



COME VINCERE LA GARA

Il vostro compito è vincere la gara.

- Per fare ciò dovrete descrivere un percorso che rispetti le regole stabilite e attraversi il **maggior numero di anelli possibile**. Il vostro punteggio sarà in corrispondenza con la percentuale di anelli attraversati.
- Il percorso deve passare in ogni quadrato *al massimo* una volta e deve essere continuo.
- Il percorso può essere sia **chiuso** (se alla fine della gara si torna al punto di partenza), sia **aperto** (se la gara si interrompe in un punto diverso da quello di partenza). I percorsi aperti sono concessi, ma con una penalità: il punteggio sarà dimezzato.

COME VINCERE LA GARA

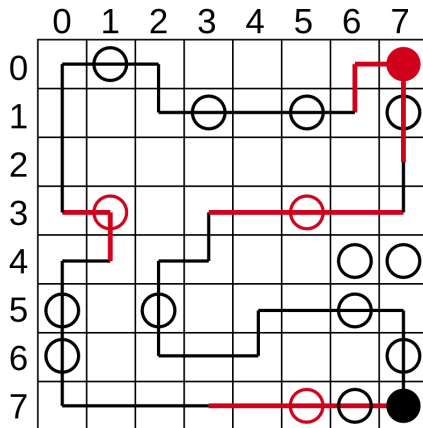
NOTA (PERCORSI APERTI)

Vi è concesso che i percorsi aperti possano iniziare o terminare in un anello. Quindi in tal caso, per questi anelli, è concesso non completare la richiesta delle regole.

NOTA (ESISTENZA DELLA SOLUZIONE)

Vi garantiamo che esiste sempre una soluzione ottima per ogni mappa, ovvero un percorso chiuso valido che attraversa tutti gli anelli.

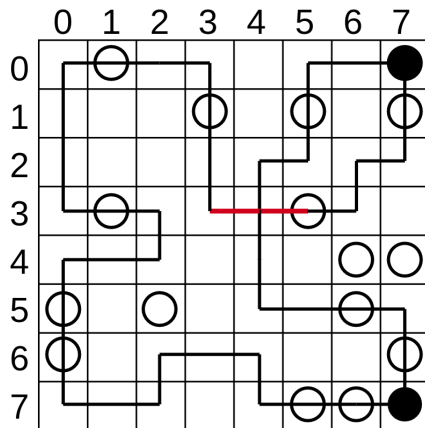
ESEMPIO - PERCORSO NON VALIDO



⇒ $B = 2, W = 15$.

⇒ **Punteggio:** 0/5 (anelli attraversati in modo invalido).

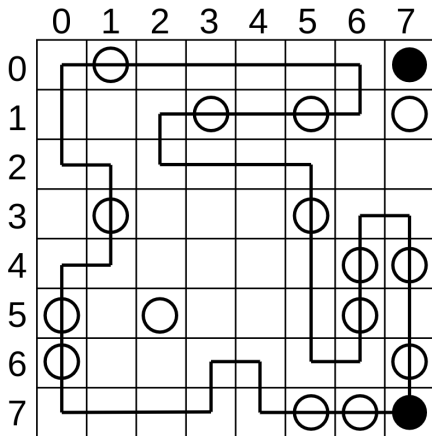
ESEMPIO - PERCORSO NON VALIDO



⇒ $B = 2, W = 15$.

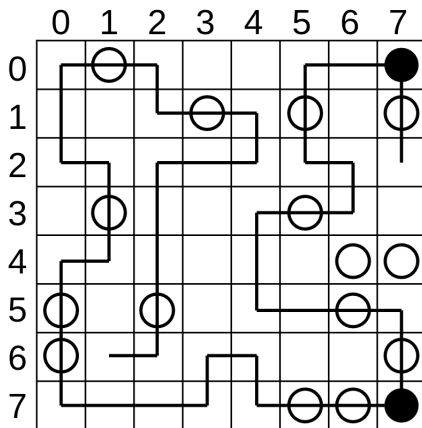
⇒ **Punteggio:** 0/5 (punto percorso due volte).

ESEMPIO - SOLUZIONE PARZIALE CHIUSA


$$\Rightarrow B = 2, W = 15.$$

⇒ **Punteggio:** 4.11/5 (anelli attraversati: 14, anelli totali: 17).

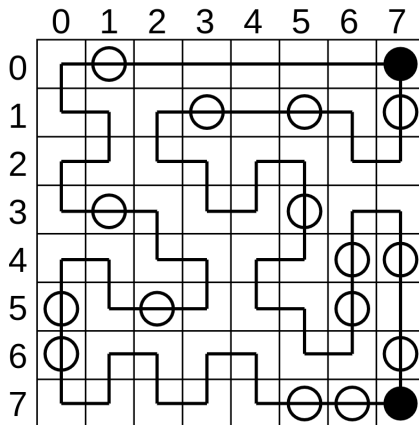
ESEMPIO - SOLUZIONE PARZIALE APERTA



⇒ $B = 2, W = 15$.

⇒ **Punteggio:** $2 \cdot 21/5$ (anelli attraversati: 15, anelli totali: 17).

ESEMPIO - SOLUZIONE COMPLETA CHIUSA



⇒ $B = 2, W = 15$.

⇒ **Punteggio:** 5/5 (anelli attraversati: 17, anelli totali: 17).

Input: la mappa del luogo in cui si svolgerà la gara. Il file di input è costituito da $1 + B + W$ righe.

- La prima riga riporta 4 numeri, N , M , B e W , rispettivamente il numero di righe e di colonne della mappa, il numero di anelli neri ed il numero di anelli bianchi presenti.
- Le successive B righe sono costituite da 2 interi ciascuna $r\ c$, che indicano riga e colonna delle posizioni degli anelli neri.
- Le successive W righe sono costituite da 2 interi ciascuna $r\ c$, che indicano riga e colonna delle posizioni degli anelli bianchi.

Output: le vostre proposte di percorsi sulla mappa, ogni proposta è composta da un'unica riga

$$A \ L \ r \ c \ d_1 d_2 d_3 \cdots d_L \#$$

dove:

- A è numero totale di anelli attraversati;
- L è la lunghezza totale del percorso effettuato;
- (r, c) è la posizione di partenza del percorso (riga e colonna);
- $d_1 \cdots d_L$ è una stringa di lunghezza L che identifica il percorso. L' i -esimo carattere della stringa fornisce la direzione in cui muoversi all' i -esimo step. Il carattere deve essere uno tra U, D, L, R (up, down, left, right);
- Ogni soluzione deve essere terminata da un cancelletto $\#$.

ESEMPIO - INPUT

INPUT

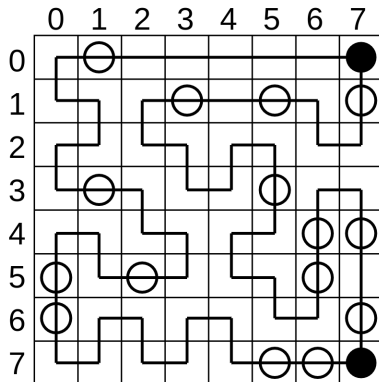
8	8	2	15
0	7		
7	7		
7	5		
1	7		
1	3		
3	5		
6	0		
5	0		
6	7		
0	1		
5	2		
4	7		
3	1		
5	6		
7	6		
4	6		
1	5		

	0	1	2	3	4	5	6	7
0		○						●
1				○		○		○
2								
3		○				○		
4							○	○
5	○		○				○	
6	○							○
7						○	○	●

ESEMPIO - OUTPUT

OUTPUT

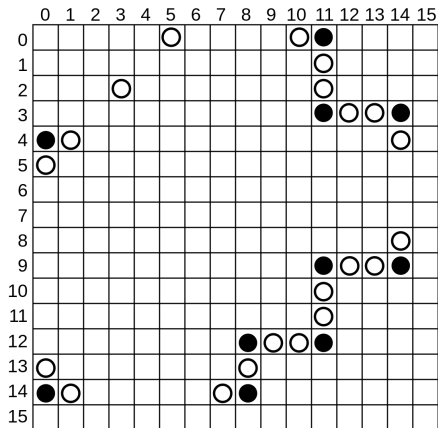
17 64 0 0 RRRRRRRDDLULLLLDRDRURDDLDRDRUUURDDDDLLLULDLULDLUUURRRULULLURULU#



ESEMPIO - INPUT

INPUT

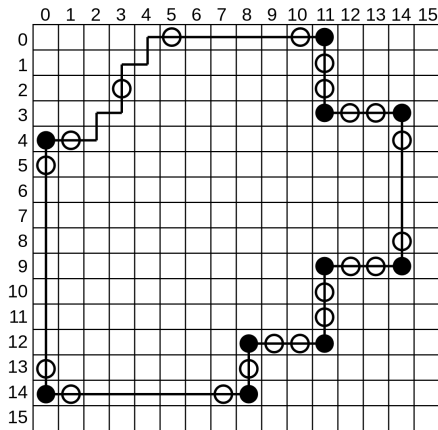
16	16	10	21	12	9
9	14			8	14
3	11			14	1
0	11			9	12
9	11			5	0
14	0			12	10
12	11			4	1
12	8			3	13
14	8			4	14
3	14			2	3
4	0			0	5
10	11			9	13
11	11			3	12
0	10			2	11
13	8			14	7
1	11			13	0



ESEMPIO - OUTPUT

OUTPUT

```
31 56 0 4 RRRRRRRDDDRRRDDDDDDLLDDDLLDDLLLLLLLLUUUUUUUUUURRURUURU#
```



ESEMPIO - INPUT

INPUT

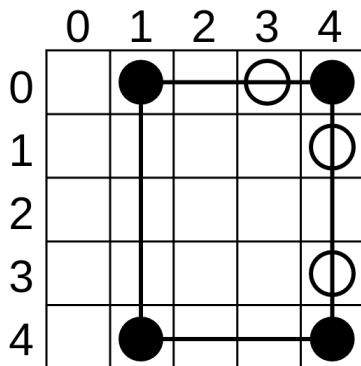
5 5 4 3
0 1
4 4
4 1
0 4
0 3
3 4
1 4

	0	1	2	3	4
0		●		○	●
1					○
2					
3					○
4		●			●

ESEMPIO - OUTPUT

OUTPUT

7 14 0 1 DDDDRRRUUUULLL#



NOTE SU INPUT

ASSUNZIONI GENERALI

- $1 < N, M \leq 256$
- $1 < B, W \leq 8000$
- $0 \leq r < N, 0 \leq c < M$, per ogni posizione (r, c) di un anello.

CASI DI TEST

- Ci sono 20 casi di test in totale.
- In almeno 6 casi su 20 esiste un percorso rettangolare come soluzione ottima.
- In almeno 10 casi su 20 vale che $1 < B, W \leq 20$.

LIMITI DELLE RISORSE

- tempo di esecuzione: 3 secondi
- memoria: 8 MB

OUTPUT

Se scrivete una soluzione esponenziale:

- Importate `swrace.h` (scaricabile da judge).
- Man mano che migliorate la soluzione, scrivetela in output terminando la riga con `#`.
- La libreria arresterà il programma prima del timeout.

```
... include delle librerie di sistema ...  
#include "swrace.h"  
int main() {  
    ...  
}
```

Note:

- Il `main` va sempre dichiarato come `int main()` o `int main(void)`.
- Il correttore considererà l'ultima riga di output che finisce con `#` quindi, anche se non stampate soluzioni multiple, terminate l'output con `#`.

COMPILARE IN LOCALE

Per testare le vostre soluzioni in locale (supponiamo che il vostro file si chiami `swrace.cpp`):

- Scaricate `grader.cpp`
- Il comando di compilazione è il seguente

```
/usr/bin/g++ -DEVAL -std=c++11 -O2 -pipe -static -s -o  
swrace grader.cpp swrace.cpp
```

I file `swrace.cpp`, `grader.cpp` e `swrace.h` devono essere nella stessa cartella.

Per sistemi Mac OS X e Windows vedere la nota nel testo.

NOTA

Per questo esercizio è necessario usare il C++, non è possibile usare il C.

PUNTEGGIO

- Ci sono 20 casi di test: ogni test assegna un punteggio di massimo 5 punti.
- Un percorso proposto è valido se rispetta tutte le regole richieste. Percorsi non validi fanno **zero punti**!
- I percorsi validi **chiusi** ottengono questo punteggio:

$$5 \cdot \left(\frac{A}{B + W} \right),$$

dove A è il numero di anelli attraversati.

- I percorsi validi **non chiusi** ottengono questo punteggio:

$$5 \cdot \left(\frac{A}{(B + W)} \right) \cdot \frac{1}{2},$$

dove A è il numero di anelli attraversati.

- Per ogni file di output viene calcolato il punteggio dato dall'**ultimo** percorso presente nel file che termina con un cancelletto #.

La sufficienza è posta a **30 punti**.

C'è un limite di **80 sottoposizioni** per gruppo, il limite è stato aumentato rispetto a quello usuale (40) perché l'interfaccia di testing non è disponibile.

DATASET DI ESEMPIO

Attenzione! Per gli input forniti nel dataset di esempio non è stata calcolata una soluzione ottima. Per questo motivo il dataset non contiene anche i relativi output, solitamente messi a disposizione.

L'assegnazione punti avviene in maniera competitiva:

- **3 punti** ai gruppi nel primo terzile della classifica (primo terzo della classifica, punteggio maggiore a quello fatto da almeno 2/3 dei gruppi);
- **2 punti** ai gruppi nel secondo terzile della classifica (secondo terzo della classifica);
- **1 punto** ai gruppi nel terzo terzile della classifica (ultimo terzo della classifica).

Vengono considerati nella classifica per l'assegnazione dei punti solamente i **gruppi che raggiungono la sufficienza** (punteggio maggiore o uguale a 30).

CONSEGNA

Consegna: 25 maggio 2020 ore 21:00

Per caricare il vostro codice, recatevi su
<https://judge.science.unitn.it/arena>

SUGGERIMENTI

Cominciate subito a lavorare al progetto per presentarvi al prossimo ricevimento (20 maggio 2020) con tutte le domande che vorrete fare. In ogni caso, risponderemo alle vostre mail.

È PERMESSO:

- 1 Discutere all'interno del gruppo
- 2 Chiedere chiarimenti sul testo
- 3 Chiedere opinioni su soluzioni
- 4 Sfruttare codice fornito nei laboratori
- 5 Utilizzare pseudocodice da libri o Wikipedia
- 6 Richiedere aiuto (anche pesante) per la soluzione "minima"
- 7 Venire a ricevimento

È VIETATO:

- ❶ Discutere con altri gruppi
- ❷ Mettere il proprio codice su repository pubblici
- ❸ Utilizzare codice scritto da altri
- ❹ Condividere codice (abbiamo potenti mezzi!)

DATE E ORARI

- mercoledì 20 maggio 2020 dalle 15:30 alle 17:30;
- venerdì 22 maggio 2020 dalle 13:30 alle 15:30;
- sabato 23 maggio 2020 dalle 10:30 alle 12:30.

⇒ Prima di venire a ricevimento (escludendo quelli in laboratorio), è obbligatorio richiedere appuntamento via mail
(bucchiarone@fbk.eu) e
(marta.fornasier@studenti.unitn.it).