

Progetto

Corso di Algoritmi e Strutture dati - A.A. 2015/2016
(1° appello)

Obiettivo del progetto è la soluzione del problema di seguito descritto utilizzando il sistema di valutazione online disponibile all'indirizzo <https://prototypes.mat.unical.it/asd>. Per la creazione dell'account necessario ad accedere al sistema, inserire i propri dati nel form¹ raggiungibile dal sito del corso e inviare una email di conferma all'esercitatore.

Problema

Durante l'ultima guerra, il grande condottiero di battaglia Cid Highwind ha lasciato alcuni tesori nelle isole di Nibelheim, segnando su delle mappe l'esatta locazione di ogni tesoro e di tutti gli ostacoli presenti. La guerra sta per finire e il nostro condottiero vuole ritirarsi a campare di rendita utilizzando i suoi tesori. Purtroppo è inseguito dalla Shinra, una organizzazione di loschi figure che lo vuole morto e quindi deve fare in fretta: deve visitare ogni isola sulla quale ha nascosto i tesori e cercare di riprenderli tutti, nel minor tempo possibile, così da non farsi acchiappare dall'organizzazione.

Ogni isola è composta da in un insieme di celle, ogni cella contiene un simbolo e ogni simbolo ha un significato diverso:

- Il simbolo **S** indica il punto in cui Cid attracca lasciando la propria nave;
- Il simbolo **-** indica dell'acqua: Cid non può camminare sulle celle che contengono acqua;
- Il simbolo **.** indica del terreno o della sabbia, che Cid può attraversare liberamente;
- Il simbolo **^** indica un gruppo di alberi: Cid non può passarci attraverso;
- Il simbolo **!** indica una torre di vedetta della Shinra: Cid deve stare lontano da esse almeno per una cella (non può passare nelle celle a destra o sinistra, sopra o sotto e le diagonali);
- Il simbolo **\$** indica un tesoro: Cid deve raccogliere tutti i tesori presenti in una mappa.

Cid attracca la sua aeronave in un punto specifico di ogni isola, e per andar via deve tornare allo stesso punto.

Sapendo che Cid può muoversi, a partire da una cella, soltanto *sopra*, *sotto*, *sinistra* o *destra*, e che per ogni movimento Cid *impiega un'ora di tempo*, si scriva un programma che presa in input la mappa, restituisca il numero minimo di ore di cui Cid ha bisogno per raccogliere tutti i tesori, partendo dal punto di attracco e ritornando allo stesso punto.

Input

La lettura dovrà avvenire da standard input.

L'input consiste in un numero **i** ($i \geq 1$) di test, rappresentanti le isole da analizzare; per ogni sessione, il formato è il seguente:

- la prima riga contiene la parola chiave **isle** e un numero **j** (separati da spazio), rappresentate l'inizio della j-esima sessione;
- la seconda riga è nel formato **n m** dove **n** rappresenta il numero di righe e **m** il numero di colonne dell'isola;
- le successive **n** righe contengono **m** simboli appartenenti all'insieme {**S**, **-**, **.**, **^**, **!**, **\$**}.

Questo formato vale per tutti i test. L'input termina con la stringa **-1**.

Si può assumere che l'input sia sempre corretto.

¹ La form è accessibile direttamente all'indirizzo: <http://bit.ly/1IZRF43>

Esempio input	Esempio output
<pre> isle 1 7 7 ----- -^\$^^^~ -...^.- --...- ---.S-- .----- ...---. isle 2 10 10 ----- --\$\$\$^^^-- -^^...^^- -^...!^^- -^\$.!!!-- --...- ---...- --...S-- -^\$.----- ----- -1 </pre>	<pre> result 1 10 result 2 32 </pre>

Output

L'output del programma deve avvenire su standard output; per ogni test, l'output deve essere nel seguente formato:

- la prima riga contiene la parola chiave **result** e un numero **k** (separati da uno spazio), rappresentanti il k-esimo test;
- la riga successiva contiene il numero minimo di ore necessarie a Cid per raccogliere tutti i tesori, partendo dal punto di attracco e ritornando allo stesso.

Regole per la consegna e la valutazione

Il sistema consente di sviluppare le proprie soluzioni sia in C++ che in Java. La compilazione delle soluzioni utilizzerà i seguenti compilatori:

- C++: g++ (Debian 4.4.5-8) 4.4.5;
- Java: vers. 1.6.0_26.

La consegna del progetto avviene mediante la sottomissione dello stesso sul sistema di valutazione e sarà considerata valida solo quando il sistema darà come risultato "CORRECT".

Fino alla scadenza fissata per ciascun appello è possibile sottomettere un qualunque numero di soluzioni. Sarà considerata ai fini della valutazione l'ultima sottomissione.

La valutazione del progetto consiste delle seguenti componenti:

- 5 punti se la soluzione fornita è corretta;
- da 0 a 6 punti attribuiti proporzionalmente all'efficienza della soluzione proposta dallo studente, rispetto all'efficienza delle soluzioni proposte da tutti gli altri studenti che partecipano al contest;
- da 0 a 6 punti relativi alla discussione del progetto in cui oltre alla padronanza del codice si valuterà la qualità della soluzione proposta (questa fase avrà luogo solo se la soluzione fornita è corretta).

Solo dopo la valutazione positiva del progetto lo studente sarà ammesso all'orale.