

LuiSS
Libera Università Internazionale
degli Studi Sociali Guido Carli

Corso di preparazione per la selezione territoriale delle Olimpiadi di Informatica

Introduzione

Giuseppe F. Italiano

23 marzo 2021

LUISS



Benvenuto a tutti!

1.053 richieste di registrazione!!!



Gestire questi numeri è una sfida: abbiamo bisogno della tua collaborazione. Etichette:

- (online) Se puoi accedere allo streaming da youtube, è meglio.
- (online) Se hai domande, ti preghiamo di farle *soltanto* nella chat di Mattermost. Non avrai altra chat al di fuori di Mattermost!!
- (online/offline) Se vuoi rispondere tu a qualche domanda sulla chat, sei il benvenuto!
- (offline) Se vuoi mandarci email (per qualcosa di veramente importante) olimpiadi@luiss.it

Benvenuto anche a docenti scuola (uditori/uditrici)

Team



Docenti



Giuseppe F. Italiano



Luigi Laura



Irene Finocchi



Nicola Prezza



Blerina Sinimeri

Tutor

- Leo Artoni
- Mattia Cervellini
- Carlo Malagnino
- Armando Pellegrini
- Emanuele Regnani

Assistenza tecnica e piattaforma gare

- Fangqing (Fred) Yuan (**袁方庆**)
- Marco Iecher
- Michele Lizzit
- Francesco Redaelli

Contatti

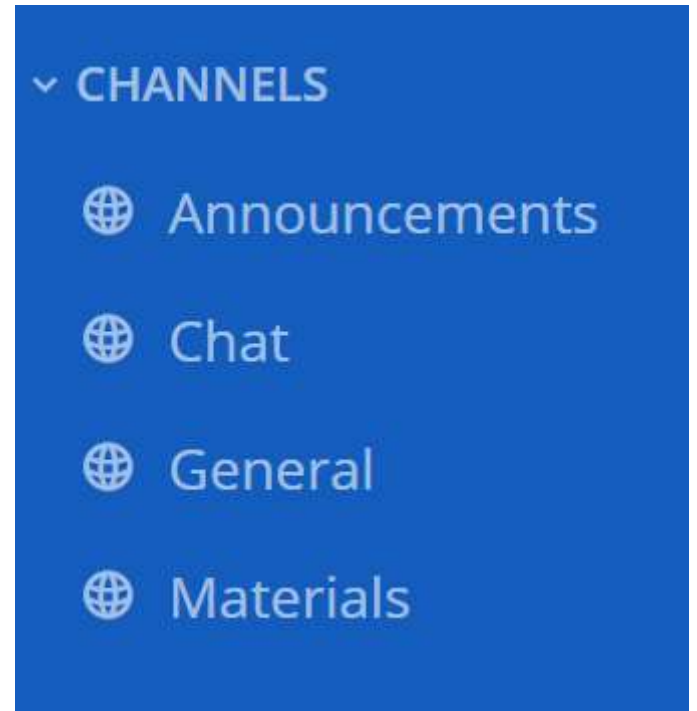
olimpiadi@luiss.it

Mattermost

<https://coding.luiss.ml/mattermost>

1. *Announcements* (e.g., link video lezioni)
2. *Chats* (fai qui le tue domande durante le lezioni)
3. *General* (discussioni interesse generale)
4. *Materials* (materiale didattico, slide etc)

Contribuisci a mantenere questi canali puliti e non postare contenuti impropri. Grazie!



Programma e calendario del corso



Ingredienti fondamentali:

1. Conoscenze di base C/C++
2. Capacità di Problem solving
3. Tecniche algoritmiche fondamentali (e.g., greedy, programmazione dinamica, grafi)

Calendario (già disponibile su Mattermost)

Data	<u>Docente</u>	Tutors	<u>Argomento</u>
25/03/2021	Blerina	Armando, Carlo, Emanuele, Leo, Mattia	C++
1/04/2021	Luigi	Armando, Carlo, Emanuele, Leo, Mattia	<u>Algoritmi e strutture dati</u>
8/04/2021	Irene + Blerina	Armando, Carlo, Emanuele, Leo, Mattia	<u>Ricorsione</u> e backtracking
15/04/2021	Irene	Armando, Carlo, Emanuele, Leo, Mattia	Greedy
22/04/2021	Giuseppe	Armando, Carlo, Emanuele, Leo, Mattia	<u>Programmazione dinamica</u>
29/04/2021	Giuseppe	Armando, Carlo, Emanuele, Leo, Mattia	Grafi
6/05/2021	Irene	Armando, Carlo, Emanuele, Leo, Mattia	Grafi
13/05/2021	Blerina + Nicola	Armando, Carlo, Emanuele, Leo, Mattia	<u>Enumerazioni, euristiche e stringhe</u>
15/05/2021	<i>GARA FINALE</i>		

Cosa sono le OII



Olimpiadi Italiane di Informatica (OII)

OII 2020/2021 (OII 2019/2020, OII 2018/2019):



- *Selezioni Scolastiche*: 10.740 (14.309, > 13.500) partecipanti
- *Selezioni Territoriali*: 2.567 (1.811, 1.358) ammessi
- *Finale Nazionale*: ??? (105, 90) ammessi

Competenze richieste

- *Selezioni Scolastiche*: Logica + saper leggere codice
- *Selezioni Territoriali*: Problem solving + tecniche algoritmiche fondamentali + saper **scrivere codice “decente”**
- *Finale Nazionale*: Tecniche algoritmiche evolute + saper scrivere buon codice

Materiale di studio



Testo

Alessandro Bugatti, “Olimpiadi di Informatica. Guida per le selezioni territoriali”. Sesta edizione, Marzo 2019.

- [https://www.imparando.net/sito/olimpiadi di informatica/guida sesta edizione.pdf](https://www.imparando.net/sito/olimpiadi%20di%20informatica/guida%20sesta%20edizione.pdf)

Molto altro materiale utile nel sito di Alessandro Bugatti:

- [https://www.imparando.net/sito/olimpiadi di informatica.htm](https://www.imparando.net/sito/olimpiadi%20di%20informatica.htm)

Messaggio importante

- Non basta seguire le nostre lezioni
- Lavora anche in autonomia, seguendo le nostre indicazioni e provando a risolvere i problemi che ti assegneremo.
- Leggi con attenzione (o meglio, studia) il testo di Bugatti
- Allenati sul portale delle olimpiadi: <https://training.olinfo.it/>
- Partecipa alla gara che organizzeremo alla fine del corso

Problem Solving



Interactive puzzles

Importante non andare a caso nella ricerca della soluzione di un problema (in gara ti penalizzerebbe!)

Collegati a <http://dm.compsciclub.ru/app/list>

Prova a risolvere tutti i problemi

Extra slides



Esempio: Scolastiche

Esercizio N° 8 – La risposta esatta vale 2 punti.

È dato il seguente programma:

```
Program E8 (input,output);
var
  mat: array[1..10,1..10,1..10] of Integer;
  i,j,k: Integer;
begin
  for i:=1 to 10 do
    for j:=1 to 10 do
      for k:=1 to 10 do
        mat[i,j,k]:=i+j+k-3;
      for i:=1 to 10 do
        write(' ',mat[i,i,i]);
      writeln()
    end.
end.
```

Cosa viene visualizzato a video dall'esecuzione del programma ?

Esempio: Territoriali

Selezioni Territoriali 2013

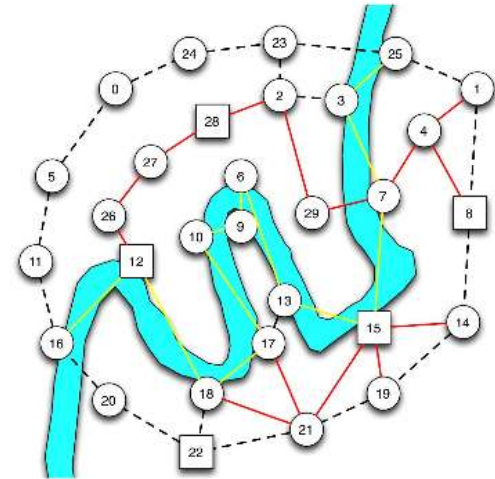
[Difficoltà D=2]

A spasso per Brisbane (brisbane)

Descrizione del problema

Nel 2013, le IOI si svolgeranno a Brisbane (in Australia). La rappresentativa italiana ha già iniziato a studiare la città, per capire cosa ci sia di interessante da vedere, e come ci si possa spostare nella giornata libera successiva alla seconda gara delle Olimpiadi. L'offerta di trasporto pubblico a Brisbane è abbastanza variegata: ci sono due linee di bus, di cui una gratuita che gira intorno alla città, e due linee di traghetti che fermano in diversi punti del fiume Brisbane, che taglia la città in due; per quello che riguarda i prezzi, esiste un abbonamento giornaliero a tutti i trasporti pubblici, bus e traghetti insieme, oppure è possibile prendere un più economico abbonamento giornaliero ai soli traghetti, o un ancor più economico abbonamento ai soli bus.

La squadra italiana vorrà visitare il maggior numero di attrazioni possibile e per questo motivo Monica, la responsabile dell'organizzazione, ha deciso di cercare un buon compromesso tra il prezzo dei biglietti e le attrazioni che sarà possibile raggiungere partendo dall'hotel. Data una lista di attrazioni e la mappa dei collegamenti delle diverse linee del trasporto pubblico, il vostro compito è quello di aiutare Monica a capire *quante attrazioni sono raggiungibili* per ogni possibile scelta dei biglietti per i trasporti pubblici.



Esempio: Territoriali

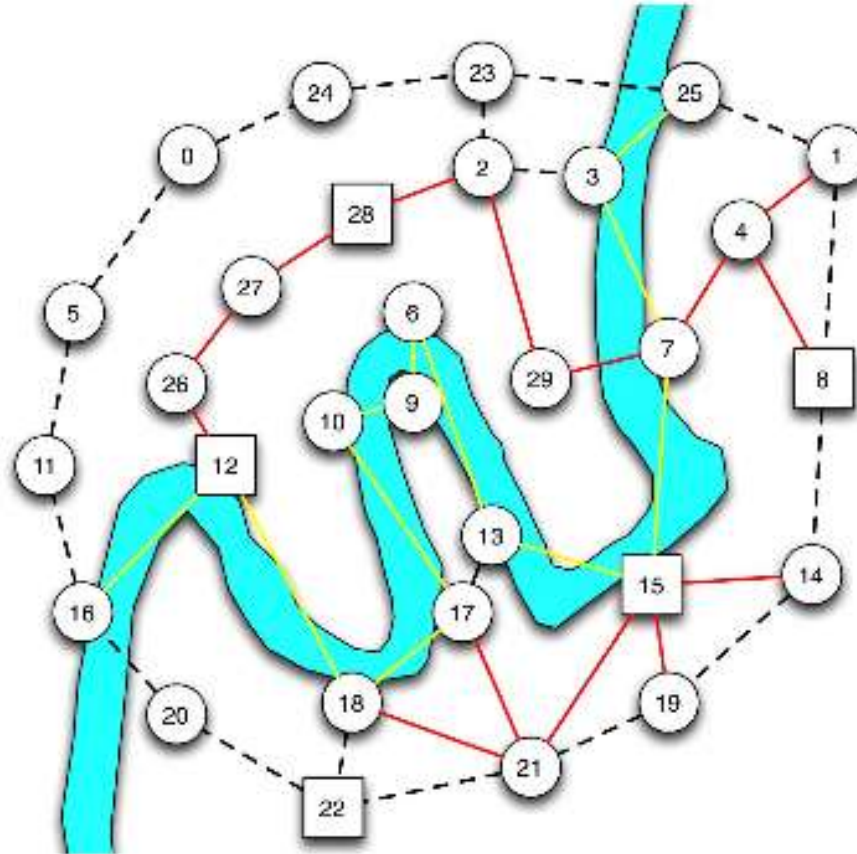
Selezioni Territoriali 2013

[Difficoltà D=2]

A spasso per Bristol

Descrizione del problema

Nel 2013, le IOI si svolgeranno a Brisbane. Studiare la città, per capire cosa ci sia di più interessante, è una giornata libera successiva alla seconda giornata di lavoro. Brisbane è abbastanza variegata: ci sono due linee di traghetti che ferma la città, e due linee di traghetti che ferma due; per quello che riguarda i prezzi, e bus e traghetti insieme, oppure è possibile solo traghetti, o un ancor più economico. La squadra italiana vorrà visitare il mare. Monica, la responsabile dell'organizzazione, ha il compito di studiare il prezzo dei biglietti e le attrazioni che saranno visitate. La squadra italiana vorrà anche la mappa dei collegamenti e quello di aiutare Monica a capire quanto costano i biglietti per i trasporti pubblici.



Esempio: Nazionali 2012



I problemi di questa edizione erano un omaggio ad Alan Turing, nel centenario della sua nascita.

Olimpiadi italiane di informatica 2012 - OII 2012

Entscheidungsproblem, o problema della fermata (fermata)

Descrizione del problema

Nota storica: nel suo famoso articolo del 1937, On computable numbers, with an application to the Entscheidungsproblem, Alan Turing dimostrò che il problema della fermata non è decidibile: tra le conseguenze, quindi, il fatto che non è possibile scrivere un programma che decida se una macchina di Turing si arresta, dato un particolare input.

Turing però è convinto che il problema della fermata sia decidibile nel modello di seguito descritto, dove si utilizza una macchina di Turing di sola lettura. La macchina ha un nastro di N celle, numerate da 0 a $N - 1$, da sinistra verso destra. In ogni cella c'è un numero intero, e le celle sono di sola lettura: la macchina non può cambiare il contenuto della cella. La macchina di Turing ha una tabella di transizione, che in funzione dello stato attuale e del numero letto, cambia lo stato interno della macchina e comanda alla macchina di spostarsi di un certo numero di celle, verso destra o verso sinistra. La cella numero 0 è una cella speciale: quando la macchina di Turing arriva nella cella 0, termina la sua computazione e si ferma.



Considerate la figura: qui vedete il nastro, con 5 celle numerate da 0 a 4, contenenti interi composti tra 0 e 2, e la tabella di transizione, che in funzione dei due stati possibili della macchina (a e b) e dei tre interi letti dalla cella, riporta lo stato successivo e lo spostamento della macchina, rappresentato da interi positivi per spostamenti verso destra e interi negativi per spostamenti verso sinistra. Per esempio, supponiamo che la macchina di Turing sia inizialmente nello stato a e che parte dalla cella 1. Nella cella 1 la macchina legge l'intero 2, come si vede dalla tabella, la macchina di Turing rimane nello stato a e si sposta di una cella a destra. Finisce quindi nella cella 2, dove legge l'intero 1: a questo punto rimane nello stato a e si sposta di due celle a sinistra; raggiunge quindi la cella 0 e si ferma. Se la macchina di Turing parte, sempre nello stato a, dalla cella 2 vediamo che termina direttamente nella cella 0, fermandosi. Viceversa, se la macchina di Turing, sempre nello stato a, parte dalla cella 3 si vede che la macchina cambia stato, passando allo stato b e spostandosi di due celle all'indietro. Si ritrova quindi nella cella 1 ma qui, dalla tabella di transizione, si vede che ritorna nello stato a e si sposta di due celle in avanti, ritornando nella cella 3. Da qui continuerà a spostarsi, alternativamente, di due celle in avanti e due celle indietro, cambiando stato a ogni spostamento. Quindi, la macchina di Turing a partire da questa configurazione iniziale, NON termina. Il vostro compito è quello di aiutare Alan Turing, scrivendo un programma che, presa in ingresso la descrizione di una macchina

1

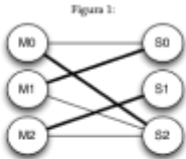
Olimpiadi italiane di informatica 2012 - OII 2012

La battaglia del convoglio (convoglio)

Descrizione del problema

Nota storica: tra il 7 e il 10 marzo del 1943 c'è stata nell'Atlantico quella che è stata definita la più grande battaglia di convogli mai combattuta. Sottilissimi tedeschi si comunicavano, in maniera cifrata, le posizioni dei convogli americani da attaccare. Gli alleati conoscevano, ovviamente, le posizioni dei loro convogli, ed intercettavano le comunicazioni dei tedeschi. Le informazioni acquisite da queste comunicazioni cifrate, insieme alle posizioni note dei convogli americani, sono state fondamentali per il lavoro di Alan Turing a Bletchley Park: qui Turing ha ideato la macchina Bomba, che ha consentito agli alleati di rompere il codice di Enigma, la macchina per comunicazioni cifrate dei tedeschi.

Retorniamo alla battaglia: un convoglio americano, composto da N navi, è in viaggio nell'Atlantico. Sottilissimi tedeschi si comunicano le posizioni delle navi e si coordinano per l'attacco. Gli alleati intercettano le comunicazioni tedesche ma riescono a decrittare solo parzialmente i messaggi: non sempre si riesce a identificare di quale nave stiamo parlando i tedeschi, e spesso più di una nave americana potrebbe essere quella a cui fanno riferimento. In particolare, se indichiamo con M_0, M_1, \dots, M_{N-1} gli N messaggi intercettati, e con S_0, S_1, \dots, S_{N-1} le N navi della flotta, alla luce di quanto decodificato ogni messaggio può riferirsi a una o più navi, come si vede nella figura (dove $N = 3$), dove, per esempio, il primo messaggio può riferirsi sia alla prima (S_0) che alla terza (S_2) nave.



Turing riesce a trovare una corrispondenza univoca tra i messaggi e le navi: una corrispondenza in cui ad ogni messaggio corrisponde una nave distinta. Per esempio, le 3 linee a tratto spesso in figura evidenziano 3 coppie messaggio-nave ($M_0 - S_2$, $M_1 - S_0$, e $M_2 - S_1$). Questa è una corrispondenza univoca in quanto:

- per ogni $i = 1, 2, 3$ esiste uno ed un solo j tale che la coppia $(M_i - S_j)$ è stata inclusa;
- per ogni $j = 1, 2, 3$ esiste uno ed un solo i tale che la coppia $(M_i - S_j)$ è stata inclusa.

Per poter proteggere la flotta bisogna essere sicuri della corrispondenza, e quindi dobbiamo ora accertarci che non esistano altre corrispondenze univoche internamente costituite da coppie messaggio-nave consentite dall'istanza (gli archi in figura, sia in grassetto che in tratto semplice). Per esempio, nel caso della Figura 1 esiste anche una seconda corrispondenza univoca: $M_0 - S_0$, $M_1 - S_2$, $M_2 - S_1$.

1

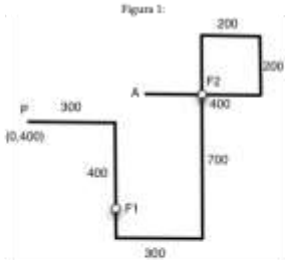
Olimpiadi italiane di informatica 2012 - OII 2012

Allenamento per la maratona (fontane)

Descrizione del problema

Nota storica: in pochi sanno che Turing era un patito maratoneta, a tal punto che il suo record personale, ottenuto il 25 agosto del 1947, 2 ore e 48 minuti e 3 secondi, è stato di soli 11 minuti superiore a quello del vincitore delle Olimpiadi del 1948 (l'argentino Delfo Cabrera, che vinse in 2 ore, 34 minuti e 31 secondi).

Alan Turing si vuole allenare per la maratona. Il suo problema è quello di rifornirsi d'acqua. Ha una mappa piuttosto accurata della zona, con segnate tutte le fontanelle disponibili, e sulla quale ha riportato il percorso che intende fare. Ha scelto un percorso formato solo da tratti in direzione orizzontale (Est-Ovest) o verticale (Nord-Sud). Turing, per serenità, consuma 1ml di acqua per ogni metro che corre: dopo aver bevuto 100ml, per esempio, è in grado di correre per 100 metri. Turing però non vuole bere mai più di 100ml per volta, e vuole correre senza essere appassito: quindi, vuole portarsi appresso una borraccia più piccola possibile. Data la mappa con segnate le fontanelle, aiutate Turing a capire qual'è la capacità della più piccola borraccia che gli consente di correre avendo sempre acqua a sufficienza.



Considerate l'esempio mostrato in figura, dove l'origine degli assi (0,0) è in basso a sinistra: qui Turing parte dal punto di coordinate (0,400) (marcato da una P) e corre lungo 7 tratti lunghi, in ordine, rispettivamente 300, 400, 300, 700, 200 e 400 metri. Ci sono due fontanelle nel percorso, la prima (marcata come F1) nel punto di coordinate (300,300) e la seconda (marcata come F2) nel punto di coordinate (400,300). Per questo percorso, Turing ha bisogno di una borraccia da 800 ml: infatti, partendo con la borraccia piena, incontra la prima fontanella dopo 600 metri. Qui Turing beve (100ml), e riempie la borraccia (800ml), così che gli fornisce l'autonomia per raggiungere la seconda

1



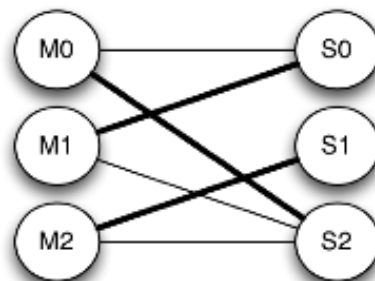
La battaglia del convoglio (convoglio)

Descrizione del problema

Nota storica: tra il 7 e il 10 marzo del 1943 c'è stata nell'Atlantico quella che è stata definita *la più grande battaglia di convogli mai combattuta*. Sottomarini tedeschi si comunicavano, in maniera cifrata, le posizioni dei convogli americani da attaccare. Gli alleati conoscevano, ovviamente, le posizioni dei loro convogli, ed intercettavano le comunicazioni dei tedeschi. Le informazioni acquisite da queste comunicazioni cifrate, insieme alle posizioni note dei convogli americani, sono state fondamentali per il lavoro di Alan Turing a Bletchley Park: qui Turing ha ideato la macchina Bomba, che ha consentito agli alleati di rompere il codice di Enigma, la macchina per comunicazioni cifrate dei tedeschi.

Torniamo alla battaglia: un convoglio americano, composto da N navi, è in viaggio nell'Atlantico. Sottomarini tedeschi si comunicano le posizioni delle navi e si coordinano per l'attacco. Gli alleati intercettano le comunicazioni tedesche ma riescono a decrittare solo parzialmente i messaggi: non sempre si riesce a identificare di quale nave stiano parlando i tedeschi, e spesso più di una nave americana potrebbe essere quella a cui fanno riferimento. In particolare, se indichiamo con $M_0..M_{N-1}$ gli N messaggi intercettati, e con $S_0..S_{N-1}$ le N navi della flotta, alla luce di quanto decodificato ogni messaggio può riferirsi a una o più navi, come si vede nella figura (dove $N = 3$), dove, per esempio, il primo messaggio può riferirsi sia alla prima (S_0) che alla terza (S_2) nave.

Figura 1:



Turing riesce a trovare una **corrispondenza univoca** tra i messaggi e le navi: una corrispondenza in cui *ad ogni messaggio distinto corrisponde una nave distinta*. Per esempio, le 3 linee a tratto spesso in figura evidenziano 3 coppie messaggio-nave ($M_0 - S_2$, $M_1 - S_0$, e $M_2 - S_1$). Questa è una corrispondenza univoca in quanto:

- per ogni $i = 1, 2, 3$ esiste uno ed un solo j tale che la coppia $M_i - S_j$ è stata inclusa;
- per ogni $j = 1, 2, 3$ esiste uno ed un solo i tale che la coppia $M_i - S_j$ è stata inclusa.