

# Prova Finale Algoritmi e Strutture Dati

## Presentazione della consegna

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)  
Politecnico di Milano

# Obiettivi didattici e realizzazione

## Obiettivi

- Applicazione pratica delle tecniche apprese nel modulo di algoritmi e strutture dati del corso di algoritmi e principi dell'informatica
- Implementazione di una soluzione ad un problema prestando attenzione ad aspetti concreti di efficienza del codice

## Realizzazione

- Linguaggio C (C11, VLA ammessi)
- Nessuna libreria esterna al di là della libreria standard C
- No multithreading
- Dati in ingresso ricevuti via `stdin`, risultati da fornire via `stdout`

# Modalità di realizzazione

- Il progetto è **strettamente individuale**
  - Non utilizzate **nulla generato da umano o macchina che non siate voi**
- Siete responsabili del vostro codice
  - Non caricatelo su repository pubblici
  - Non condividetelo con colleghi per “prendere ispirazione”
  - Non utilizzate alcun frammento di codice reperito su Internet
  - Non utilizzate assistenti automatici come Copilot, ChatGPT e simili
- In caso di plagi o uso di codice altrui, **tutti i progetti coinvolti** saranno annullati

# Criteri di valutazione

## Efficacia

- Correttezza ed efficienza della soluzione proposta sono valutate con batterie di test automatizzate
- Verranno forniti input/output d'esempio per poter collaudare la soluzione in locale
  - Non sottoponete soluzioni senza aver verificato che funzionino localmente
  - Verrà fornito anche uno strumento di generazione automatica di casi di test (input/output), per facilitarvi il testing in locale

## Efficienza

- Il sistema di verifica calcola il tempo macchina e la memoria utilizzati
- La valutazione è immediatamente calcolata (e subito visibile), mediante 6 batterie di test (task, nel lessico del verificatore):
  - Ogni batteria ha una valutazione associata tra queste: {18, 21, 24, 27, 30, 30 e lode}
  - Per ottenere una valutazione X è necessario e sufficiente superare la batteria di test con valutazione associata X

## Criteri di valutazione - 2

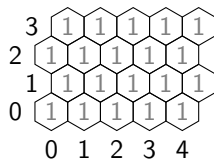
- Nessun limite al numero di sottoposizioni, né penalità per sottoposizioni multiple
- È possibile migliorare la valutazione quante volte si desidera
- Avvertenza: viene valutata l'ultima sottoposizione fatta.
  - Sottoponete sempre il vostro sorgente definitivo al test in cui ottenete la valutazione più alta
- Verificatore disponibile all'indirizzo <https://dum-e.deib.polimi.it>
- Credenziali di accesso ricevute via mail istituzionale polimi
- Invio delle credenziali e apertura verificatore entro il 4 giugno

# Scadenze e Pianificazione

- Per i laureandi di luglio:
  - 30 giugno 2025, ore 23.59 CEST. Segnalate (email al docente) la necessità di valutazione
- Per tutti gli altri:
  - 8 settembre 2025, ore 23.59 CEST, dopo di che la piattaforma verrà chiusa. Non segnalate la necessità di valutazione.
- Per laureandi di gennaio/febbraio (SUPERATI 145 CFU e iscritti all'esame di laurea):
  - la piattaforma sarà riaperta per 10 giorni nella sessione d'esame invernale (gennaio/febbraio 2026). Le date precise saranno comunicate quando verrà definito il calendario dell'AA 25-26.
- Iniziare a lavorare ad una settimana dalla scadenza è uno dei modi migliori per non riuscire a superare la prova

- Tutor suddivisi per fascia di cognome
  - Cognomi da A a Coppelli : Tommaso Felice Banfi [tommasofelice.banfi@mail.polimi.it](mailto:tommasofelice.banfi@mail.polimi.it)
  - Cognomi da Coppola a Liao : Andrea Brugnera [andrea.brugnera@mail.polimi.it](mailto:andrea.brugnera@mail.polimi.it)
  - Cognomi da Libera a Putelli : Giorgio Barocco [giorgio.barocco@mail.polimi.it](mailto:giorgio.barocco@mail.polimi.it)
  - Cognomi da Putrone a Zuo : Filiberto Canino [filiberto.canino@mail.polimi.it](mailto:filiberto.canino@mail.polimi.it)
- Gruppo Telegram gestito dai tutor: <https://t.me/tutoratoAPI>
- Incontro con descrizione degli strumenti di sviluppo tenuto da A. Barenghi
  - Aula 5.03, 2025-06-03 dalle 14:15 alle 16:15
  - Streaming: <https://politecnicomilano.webex.com/meet/alessandro.barenghi>
  - La registrazione sarà resa disponibile

- **Movhex** è una compagnia di autotrasporti che dispone di una flotta di veicoli dispersi su un'ampia area geografica.
- Il programma da realizzare calcola i costi di spostare veicoli data una mappa formata da piastrelle esagonali dell'area
- La mappa è una piastrellatura rettangolare di esagoni
- Ogni esagono è identificato univocamente dalle sue coordinate  $(x_p, y_p)$  e ha un costo di attraversamento (intero) associato

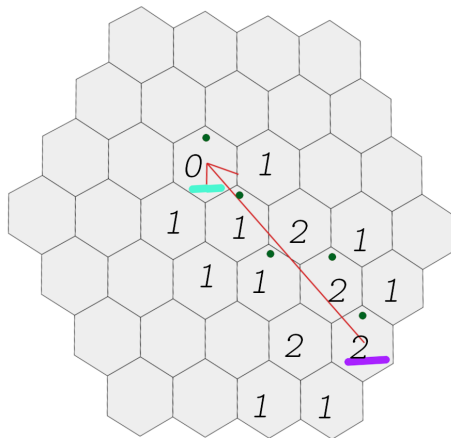


**Figure:** Una mappa con 4 righe e 5 colonne



# Costo di uno spostamento e criterio di calcolo

- Il costo di muoversi da un esagono  $(x_s, y_s)$  ad un esagono  $(x_d, y_d)$  adiacente ad esso è il costo indicato nell'esagono  $(x_s, y_s)$ .
- Esagoni con costo 0 sono intransitabili
  - n.b. un esagono a costo 0 può essere la destinazione di un percorso
- Esempio a lato:
  - Partenza: esagono con linea viola
  - Destinazione: esagono con linea azzurra
  - Il percorso migliore è indicato dai punti verdi. Il costo è  $2 + 2 + 1 + 1 = 6$ .



Comandi: `travel_cost`  $\langle x_p \rangle$   $\langle y_p \rangle$   $\langle x_d \rangle$   $\langle y_d \rangle$

- Obiettivo: il comando richiede di trovare il percorso a costo minimo per raggiungere l'esagono destinazione  $(x_d, y_d)$  da quello di partenza  $(x_p, y_p)$
- La risposta al comando è il numero intero corrispondente al costo del percorso trovato
- Casi particolari:
  - La risposta è 0 se la partenza  $(x_p, y_p)$  coincide con la destinazione  $(x_d, y_d)$
  - La risposta è  $-1$  se non è possibile raggiungere la destinazione  $(x_d, y_d)$  da  $(x_p, y_p)$

Comandi: `init  $\langle y \rangle$   $\langle x \rangle$`

- Crea una nuova mappa con  $y$  colonne e  $x$  righe, eliminando le informazioni precedenti
- Il costo di transito di ogni esagono è impostato a 1

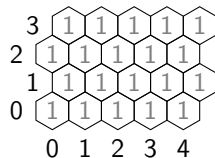


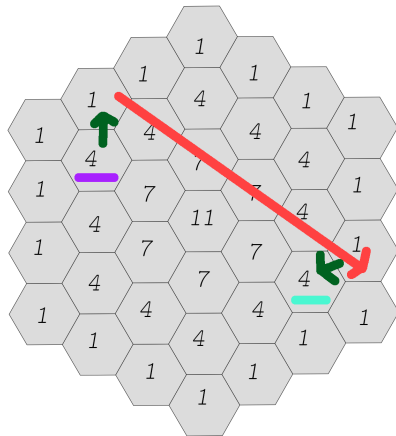
Figure: Risultato di `init 5 4`

# Rotte Aeree

- Durante l'esecuzione, gli **esagoni** possono ottenere o perdere collegamenti diretti monodirezionali tra loro tramite apertura e chiusura di **rotte aeree**
- Una rotta aerea è identificata univocamente dalla coppia ordinata dei due esagoni che essa collega, rendendoli adiacenti tra loro
- Da ogni esagono partono al più 5 **rotte aeree**
- Ogni rotta aerea è associata ad un costo di transito
  - Il **costo** della **rotta** aerea viene calcolato al momento della sua aggiunta.
  - Il **costo** della **rotta** aerea è determinato dalla media (approssimata per difetto) dei **costi** di *tutte* le connessioni aeree precedentemente esistenti uscenti dall'esagono e del costo di attraversamento dell'esagono di partenza. 
$$\text{costo} = \frac{\sum (\text{costi connessioni aeree}) + \text{costo esagono}}{|\text{connessioni aeree}| + 1}$$
- Il costo delle rotte aeree è limitato, dopo l'aggiornamento, all'intervallo tra 0 e 100 inclusi.

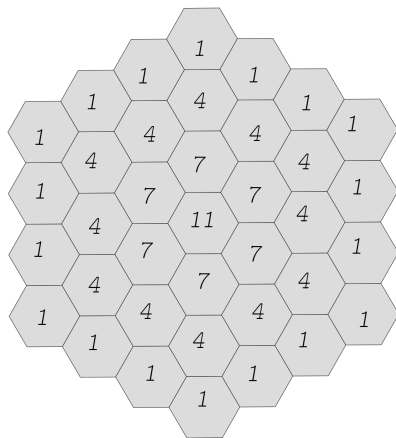
## Comandi: `toggle_air_route` $\langle x_1 \rangle$ $\langle y_1 \rangle$ $\langle x_2 \rangle$ $\langle y_2 \rangle$

- Crea (o rimuove se già presente) una connessione monodirezionale con partenza in  $(x_1, y_1)$  e arrivo in  $(x_2, y_2)$
- Esempio a lato:
  - La freccia rossa indica una rotta aerea
  - Il percorso ottimale è costituito dalla rotta aerea indicata e dai due percorsi terrestri verdi
  - Raggiungere destinazione (riga azzurra) dalla partenza (riga viola), ha costo minimo 6



## Comandi: `change_cost` $\langle x \rangle$ $\langle y \rangle$ $\langle v \rangle$ $\langle raggio \rangle$

- Modifica il costo degli esagoni in un'area circolare centrata in  $(\langle x \rangle, \langle y \rangle)$  con raggio pari a  $\langle raggio \rangle$
- Al costo di ogni esagono è sommata una frazione di  $\langle v \rangle$  inversamente proporzionale al numero di esagoni che lo separano da  $(\langle x \rangle, \langle y \rangle)$
- Il valore di  $\langle raggio \rangle$  è un intero positivo
- Il valore di  $\langle v \rangle$  è un intero
- Esempio a lato:
  - Risultato dell'applicazione di `change_cost`  $x$   $y$  10 3 ad una zona di esagoni con tutti a costo 1
  - L'esagono  $(x, y)$  è quello in posizione centrale con valore 11 post aggiornamento



## Comandi: `change_cost` - calcolo dell'incremento

- Detto  $costo_{(x_e, y_e)}$  il costo attuale dell'esagono, a valle dell'applicazione del comando `change_cost`  $\langle x \rangle$   $\langle y \rangle$   $\langle v \rangle$   $\langle raggio \rangle$  è calcolato come:

$$costo_{(x_e, y_e)} + \left\lfloor \langle v \rangle \times \max \left( 0, \left( \frac{\langle raggio \rangle - \text{DistEsagoni}((x_e, y_e), (\langle x \rangle, \langle y \rangle))}{\langle raggio \rangle} \right) \right) \right\rfloor$$

- Il costo calcolato con la formula precedente è limitato all'intervallo di interi tra 0 e 100:
  - nel caso in cui il risultato sia negativo, questo viene posto a 0
  - se il risultato eccede 100, viene impostato a 100
- **NB:** La procedura ha effetto anche sul costo di tutte le connessioni aeree uscenti dagli esagoni modificati.

## Esempio d'esecuzione

Comando	Risp.	Commento
init 100 100	OK	Inizializzazione mappa
change_cost 10 20 -10 5	OK	Rende intransitabile una regione della mappa
change_cost 30 95 10 1	OK	Aumenta il costo delle connessioni in uscita del solo (30, 95)
travel_cost 0 0 20 0	20	Costo minimo del viaggio
travel_cost 30 95 30 97	12	Costo di attraversare due connessioni, di cui una modificata da change_cost
travel_cost 10 20 11 20	-1	Destinazione non raggiungibile da (10, 20)
toggle_air_route 0 0 20 0	OK	Aggiunge rotta tra (0, 0) e (20, 0)
travel_cost 0 0 20 0	1	Ora il costo tra le due caselle è 1 grazie alla nuova connessione
toggle_air_route 10 20 10 22	OK	Connette un esagono prima irraggiungibile; la media dei costi è 0 → resta intransitabile



## Esempio d'esecuzione – segue dalla precedente

Comando	Risp.	Commento
travel_cost 10 20 10 22	-1	Sia la connessione per adiacenza sia quella addizionale sono intransitabili
toggle_air_route 0 0 20 0	OK	Rimuove rotta tra (0,0) e (20,0)
travel_cost 0 0 20 0	20	Rimossa la connessione addizionale, il costo torna quello delle connessioni intermedie
change_cost 200 20 -10 5	K0	Un argomento indica un esagono inesistente
toggle_air_route 200 20 -10 5	K0	Un argomento indica un esagono inesistente
travel_cost 200 20 11 20	-1	Un argomento indica un esagono inesistente
init 20 20	OK	La mappa è reinizializzata
change_cost 10 20 -10 5	K0	Dopo la reinizializzazione (10, 20) non è più una cella valida