

# Caso di studio di Ingegneria della Conoscenza

Membri:

Francioso Donato (661621)

[d.francioso7@studenti.uniba.it](mailto:d.francioso7@studenti.uniba.it)

Zaccaria Alessio Antonio (661060)

[a.zaccaria31@studenti.uniba.it](mailto:a.zaccaria31@studenti.uniba.it)

## INTRODUZIONE

Nel caso di studio si andrà ad affrontare la ricerca per il cammino minimo in un grafo con l'algoritmo A\* e la gestione di risultati tramite dei valori inseriti da input con il linguaggio Datalog.

Il tutto per proporre ad un nuovo visitatore un itinerario personalizzato secondo degli attributi, come ad esempio la presenza di bambini o la durata dell'attrazione.

## IL PARCO GIOCHI:

Il nostro parco giochi è un grafo pesato i cui nodi compongono le attrazioni e gli archi i percorsi che li collegano. Ogni arco ha un suo peso che corrisponde ai metri necessari per arrivare da un'attrazione alla successiva.

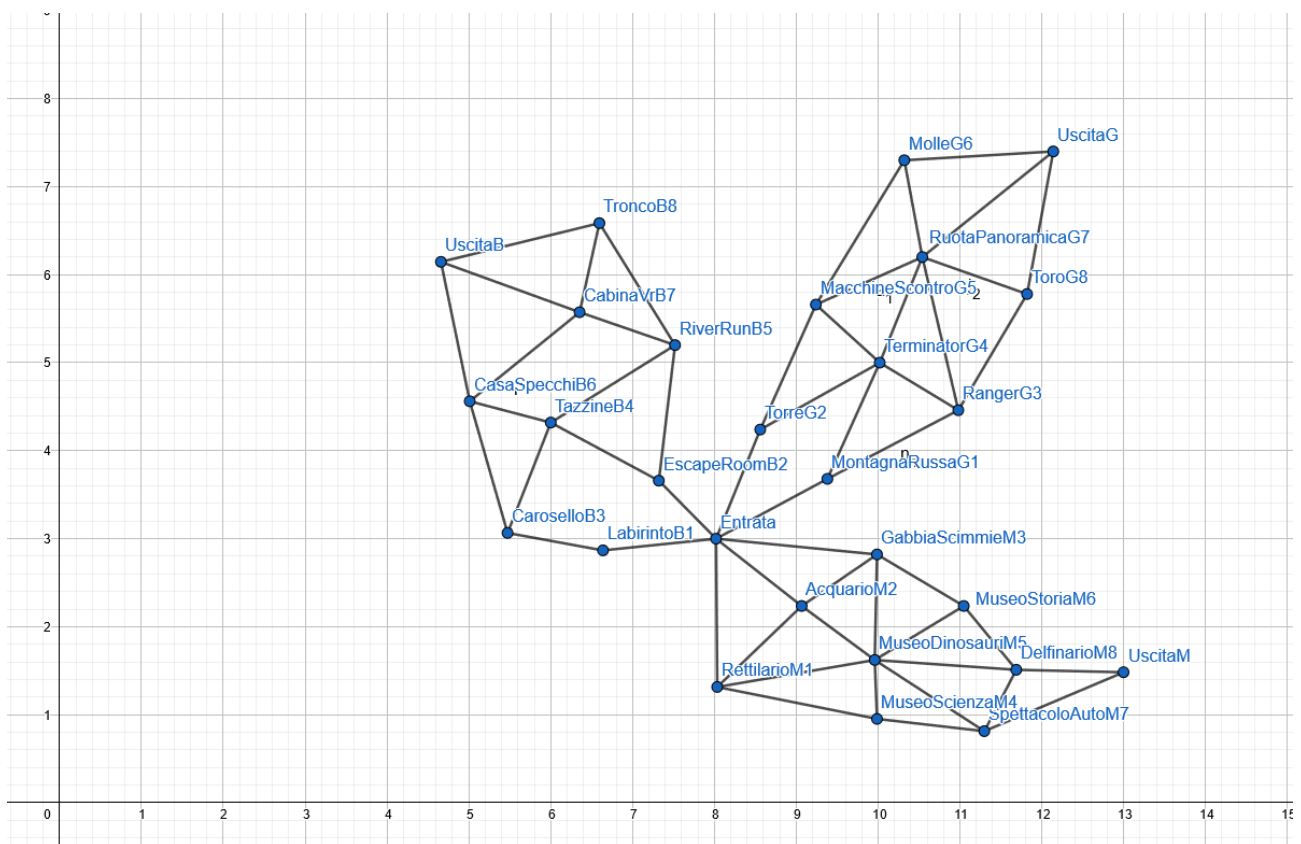


Immagine del parco giochi su piano cartesiano

## ATTRAZIONE:

L'attrazione è composta da un nome unico, dal numero di posti che l'attrazione possiede, dal tempo necessario per completarla, da una coppia di numeri che corrispondono alla loro posizione sul piano cartesiano, da un dato variabile che corrisponde alla fila presente.

Per ogni attrazione viene poi calcolata una variabile chiamata da noi TAS che corrisponde al tempo di attesa, in base ai posti e alla fila, necessario per poterne usufruire.

$$TAS = \left( \left( \frac{\text{fila}}{\text{posti}} \right) * \text{tempo} \right) + \text{tempo}$$

**PROBLEMA:** Si vuole stampare a schermo il cammino più breve per completare la sezione del parco. Non si vuole solo calcolare la distanza ma anche il tempo di attesa ad un'attrazione.

### CAMMINO MINIMO

Per cercare il cammino più veloce da percorrere per arrivare all'uscita si è deciso di utilizzare l'algoritmo di ricerca su grafi A\*.

A\* sceglie il percorso migliore utilizzando una stima dal nodo preso in considerazione calcolato come il costo del nodo corrente dal nodo iniziale sommato a una stima euristica dal nodo corrente al nodo finale.

Nel nostro caso il costo base dal nodo iniziale veniva calcolato non solo sommando il peso degli archi ma anche il TAS delle attrazioni.

La stima invece è stata calcolata sommando una distanza in linea retta dal nodo corrente all'uscita (utilizzando il teorema di Pitagora) con il tempo medio di completamento di un'attrazione generica considerando che la distanza massima tra due attrazioni non sia superiore ai 20 metri.

Nello svolgimento quindi utilizziamo:

- Una coda prioritaria chiamata "open" dove andiamo ad inserire tutti i nodi che incontriamo per la prima volta insieme al loro costo (costo base + stima).
- Una lista chiamata "close" in cui andiamo ad inserire tutti i nodi già controllati.
- Un dizionario chiamato "came\_from" per salvare il percorso finale.

Nello specifico si inserisce il nodo entrata all'interno di open, si controllano i nodi collegati, si inseriscono in open se non sono già presenti o si aggiorna il loro valore nel caso sia minore a quello già esistente. Si aggiunge il nodo corrente alla lista close e in came\_from si aggiunge il nodo padre del nodo corrente. Questo finché dalla lista di open viene prelevato il nodo Uscita. In quel caso il programma si ferma e stampa a ritroso da came\_from tutti i nodi utilizzati per raggiungere l'uscita dall'entrata.

**PROBLEMA:** Visualizzare delle attrazioni diverse in base alle esigenze di un visitatore.

### SUGGERIMENTI AL CLIENTE

Per i suggerimenti delle attrazioni utili in base alle esigenze del cliente abbiamo utilizzato il linguaggio di interrogazione Datalog. Abbiamo definito degli atomi per tutte le attrazioni e chiediamo al cliente di rispondere a delle domande da noi scelte. In seguito abbiamo creato delle clausole per ogni sezione che restituiscono un risultato diverso in base alle esigenze inserite dal cliente. Così facendo potrà vedere quali sono le attrazioni adatte. Potrà comunque scegliere la sezione desiderata per poi andare ad eseguire A\*.

## ESEMPIO DI ESECUZIONE

Per eseguire il programma si dovrà installare pyDatalog con il comando “pip install pyDatalog”.

Una volta eseguito avremo a video la lista di tutte le attrazioni disponibili nel parco giochi.

```
Buongiorno e benvenuto al parco divertimenti, di seguito una lista di tutte le giostre divise nelle tre sezioni!  
Sezione G: ['Labirinto', 'Escape Room', 'Carosello', 'Tazzine', 'River run', 'Casa delgi specchi', 'Cabina VR', 'Tronco  
']  
  
Sezione B: ['Torre', 'Montagna Russa', 'Ranger', 'Terminator', 'Macchine da scontro', 'Molle', 'Ruota Panoramica', 'Toro  
']  
  
Sezione M: ['Retttilario', 'Acquario', 'Gabbia delle scimmie', 'Museo scienze', 'Museo dinosauri', 'Museo storia', 'Spett  
acolo auto', 'Delfinario']
```

Dopodiché ci verranno poste delle domande a cui dovremo rispondere con delle parole specifiche.

Alla fine di queste domande verranno visualizzate a schermo le attrazioni suggerite in base alle nostre esigenze e verrà chiesto quale sezione scegliere. In base alle scelte effettuate viene visualizzato il cammino più veloce per completare la sezione.

Di seguito due esempi:

### Esempio 1

```
Avete bambini? no  
  
Inserisci la preferenza della durata tra corta/lunga/media: corta  
  
Inserisci che emozione vuoi tra tranquillo/adrenalinico: tranquillo  
  
Nessuna giostra corrisponde alle sue esigenze nel settore G  
  
Settore B:  
Utente  
-----  
Tazzine  
River run  
Carosello  
Cabina VR  
  
Settore M:  
Nessuna giostra corrisponde alle sue esigenze nel settore M  
  
Inserire la scelta della sezione desiderata tra g/b/m:b  
  
Riguardo al settore scelto consigliamo il percorso più breve da eseguire in base all'attuale coda nelle attrazioni:  
  
Uscita  
Casa delgi specchi  
Tazzine  
Escape Room  
Entrata
```

## Esempio 2

Avete bambini? si

Inserisci la preferenza della durata tra corta/lunga/media: lunga

Inserisci che tipo di feedback vuoi tra attivo/passivo: passivo

Nessuna giostra corrisponde alle sue esigenze nel settore G

Nessuna giostra corrisponde alle sue esigenze nel settore B

Utente

-----  
Museo scienze  
Museo dinosauri  
Museo storia  
Delfinario  
Spettacolo auto

---

Inserire la scelta della sezione desiderata tra g/b/m:m

Riguardo al settore scelto consigliamo il percorso più breve da eseguire in base all'attuale coda nelle attrazioni:

Uscita  
Delfinario  
Museo dinosauri  
Acquario  
Entrata

## DUE ESEMPI PER A\*

In questo primo caso la coda (quarto parametro) è minore in m1 rispetto ad m2 e in m7 rispetto a m8 e fornisce questo risultato

Uscita  
Spettacolo auto  
Museo scienze  
Retttilario  
Entrata

```
m1 = Giostra("Retttilario", 20, 10, 30, pm1, 0) #a1
m2 = Giostra("Acquario", 20, 10, 80, pm2, 0) #b1
m3 = Giostra("Gabbia delle scimmie", 20, 10, 80, pm4, 0) #c1
m4 = Giostra("Museo scienze", 50, 20, 5, pm3, 0) #d1
m5 = Giostra("Museo dinosauri", 50, 20, 70, pm5, 0) #u
m6 = Giostra("Museo storia", 50, 20, 10, pm6, 0) #t
m7 = Giostra("Spettacolo auto", 100, 25, 30, pm7, 0) #v
m8 = Giostra("Delfinario", 100, 25, 80, pm8, 0) #w
muscita = Giostra("Uscita", 1, 1, 0, pmuscita, 0)
```

Al contrario invertendo le maggioranze abbiamo un percorso diverso.

Uscita  
Delfinario  
Museo dinosauri  
Acquario  
Entrata

```
m1 = Giostra("Rettilario", 20, 10, 30, pm1, 0) #a1
m2 = Giostra("Acquario", 20, 10, 20, pm2, 0) #b1
m3 = Giostra("Gabbia delle scimmie", 20, 10, 80, pm4, 0) #c1
m4 = Giostra("Museo scienze", 50, 20, 15, pm3, 0) #d1
m5 = Giostra("Museo dinosauri", 50, 20, 60, pm5, 0) #u
m6 = Giostra("Museo storia", 50, 20, 13, pm6, 0) #t
m7 = Giostra("Spettacolo| auto", 100, 25, 90, pm7, 0) #v
m8 = Giostra("Delfinario", 100, 25, 10, pm8, 0) #w
muscita = Giostra("Uscita", 1, 1, 0, pmuscita, 0)
```