

Programma Arnaldo 2019

Esercizio 6.2

Le Rovine Perdute



Lezione 6.2

Le Rovine Perdute

Sulle cime delle Ande, fra foreste incontaminate dove l'uomo da secoli non osa avventurarsi, si celano le **Rovine Perdute**: i resti di un'antica e ormai scomparsa civiltà, oggetto di desiderio da parte di tutti gli esploratori che vogliano definirsi tali.

Un gruppo di archeologi, suddiviso in due squadre di esplorazione, ha deciso di avventurarsi per la giungla convinto di aver finalmente identificato le coordinate geografiche della leggendaria città: attraversando i vecchi insediamenti di quell'antico popolo e ricostruendo le tracce che hanno lasciato, sono sicuri di poter raggiungere la meta tanto ambita.

La prima squadra, denominata **Team Tonatiuh**, si muoverà per i sentieri di quelle valli con un avveniristico veicolo progettato per affrontare ogni tipo di dislivello senza alcuna difficoltà. Al contrario, il **Team Metztli** sfrutterà le potenzialità di un veicolo in grado di viaggiare a costo pressoché nullo su traiettorie piane.

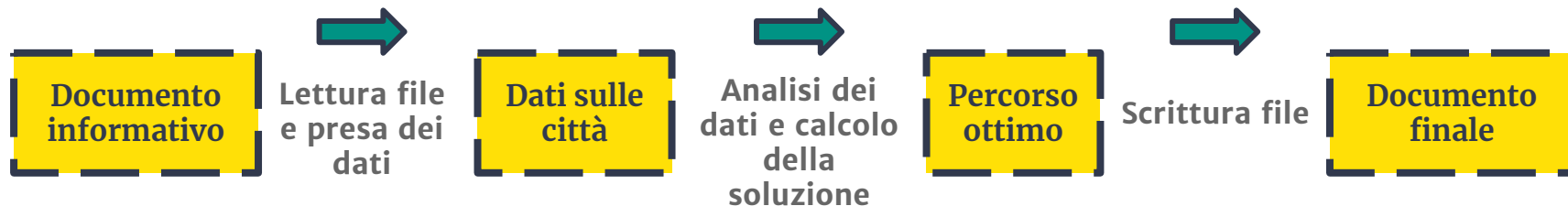
Entrambi i team sono convinti della superiorità dei propri veicoli, ma chi fra i due avrà operato la scelta vincente?

Obiettivo del programma

Il vostro **Team di Programmatori** è stato assoldato dalle due squadre di esploratori che stanno per partire alla ricerca delle meravigliose e leggendarie *Rovine Perdute*.

Vi verranno forniti tutti i documenti reperibili sulla zona, frutto di anni di studi archeologici e geografici, nei quali potrete trovare tutte le informazioni relative ai sentieri e ai percorsi che si intrecciano sulle alture circostanti, collegando città abbandonate di un'antica popolazione ormai scomparsa.

Il vostro obiettivo sarà quello di calcolare il **percorso ottimo** per entrambi i team di esplorazione, secondo le specifiche richieste di ciascuno di loro, e di produrre un documento a vostra volta indicando la sequenza di città da attraversare e il costo complessivo dei due percorsi.



Le città

Il percorso per le *Rovine Perdute* si snoda sulle montagne attraversando vari resti di antichi insediamenti. Questi insediamenti, un tempo floride e splendenti città, sono caratterizzati da:

- Due coordinate cartesiane, che indicano rispettivamente l'**ascissa** (X) e l'**ordinata** (Y) della città sulla mappa.
- Una terza quantità indicante l'**altitudine** (H) della città, ossia i metri sul livello del mare.
- Inoltre, ogni città è conosciuta con un **nome** in linguaggio comune, dato dai primi esploratori della zona o adattato dalle tradizioni dalle popolazioni autoctone.
- Infine, ogni città viene identificata dagli archeologi grazie ad un **intero univocamente assegnato** (ID).

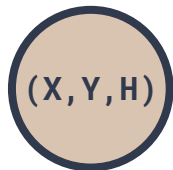
Da ogni città, inoltre, partono dei percorsi che si connettono alle città vicine, strade e mulattiere più o meno strette che permettono i trasporti in queste zone ancora selvagge.

Gli esploratori sanno bene che non è saggio avventurarsi al di fuori dei sentieri segnalati; pertanto, potete considerare di utilizzare **solo i collegamenti prestabiliti** fra due città, senza tracciarne di nuovi.

Esempio 6.2.1

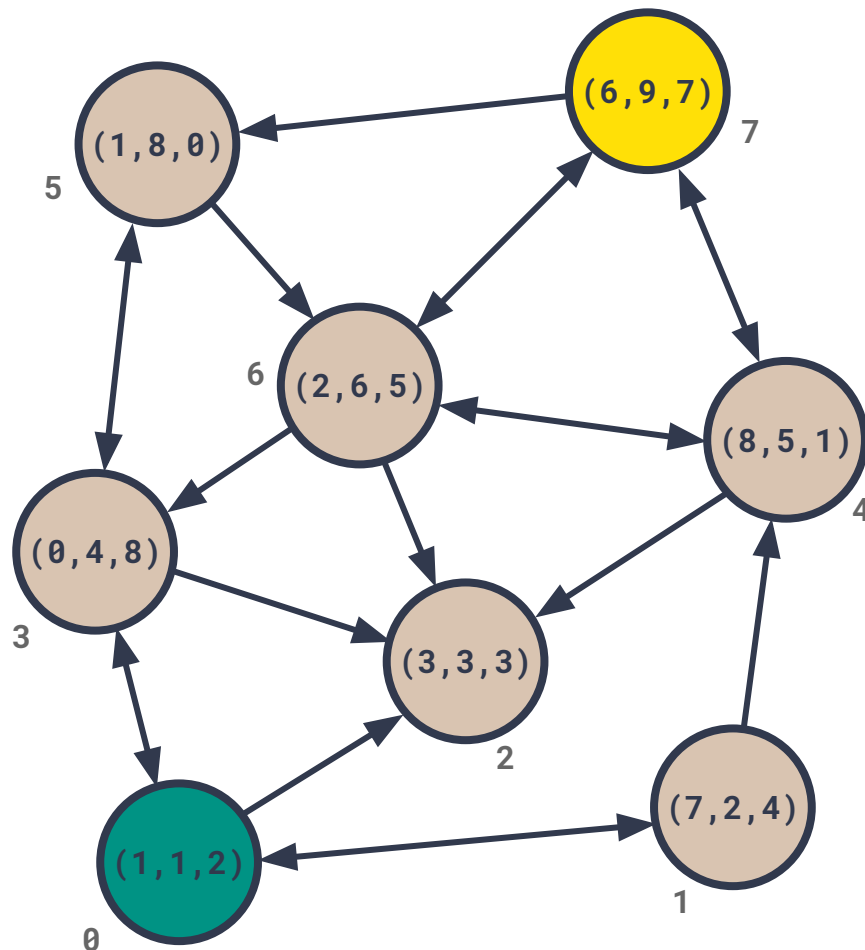
Esempio di una **mappa** con 8 città (di cui una è il campo base e un'altra sono le leggendarie *Rovine Perdute*).

Legenda:



X = ascissa;
Y = ordinata;
H = altitudine;

-  = Campo base
-  = Rovine Perdute



Il documento informativo

Il documento informativo consiste in un file XML, all'interno del quale sono indicate le caratteristiche di ogni singola città e i relativi collegamenti, secondo la seguente struttura:

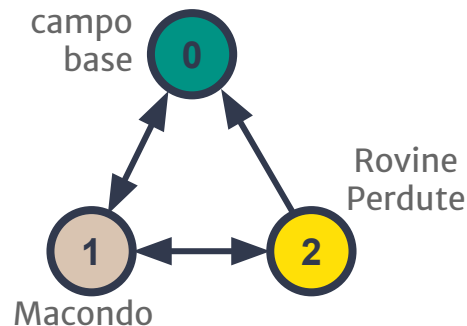
```
<map size="numero_di_città">
  <city id="numero_univoco" name="nome_città" x="." y="." h=".">
    <link to="id_altra_città"> // Collegamento monodirezionale!
    <link to="...">
    ...
  </city>
  <city ...>
    ...
  </city> // Nota: le Rovine Perdute sono sempre la città con ID maggiore!
  ...
</map>
```

Esempio 6.2.2

Un esempio di corpo di **documento informativo** costituito da tre città (*campo base*, *Macondo* e le *Rovine Perdute*) collegate fra loro:

```
<map size="3">
  <city id="0" name="campo base" x="0" y="0" h="2">
    <link to="1">
  </city>
  <city id="1" name="Macondo" x="3" y="4" h="9">
    <link to="0">
    <link to="2">
  </city>
  <city id="2" name="Rovine Perdute" x="8" y="16" h="11">
    <link to="0">
    <link to="1">
  </city>
</map>
```

Schema dei collegamenti:



L'obiettivo dei veicoli

Dopo aver analizzato il documento informativo di cui disponete, potete passare all'elaborazione della richiesta degli esploratori.

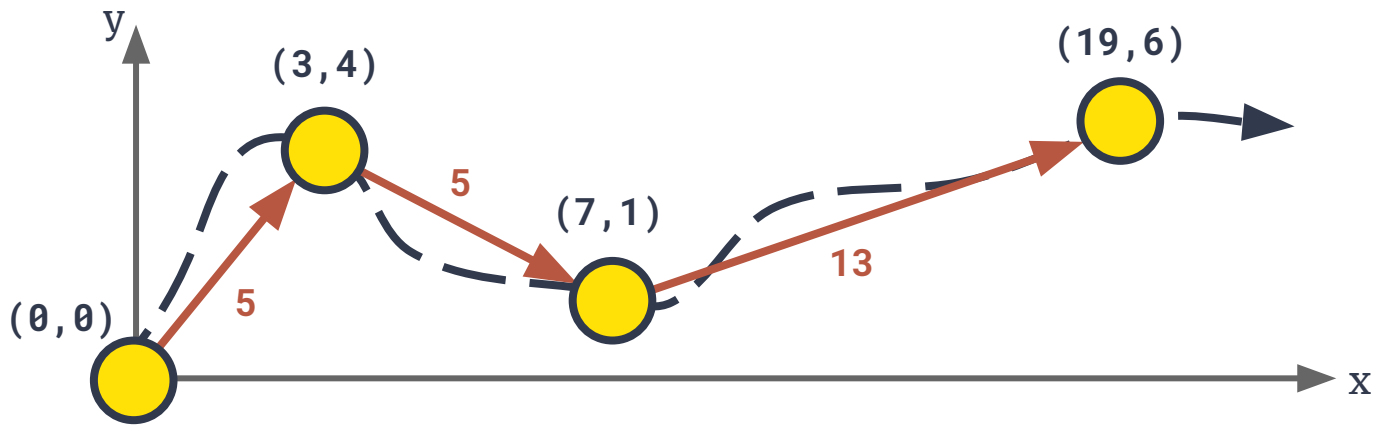
I due team, sebbene abbiano progettato veicoli completamente diversi, sono accomunati dallo stesso obiettivo: **minimizzare la quantità di carburante richiesta**, calcolata sulla base del funzionamento del rispettivo veicolo (i dettagli sono forniti nelle pagine seguenti). Il vostro compito, quindi, consiste nel trovare il **percorso ottimo** che permetta ai due team di consumare la quantità minima di carburante.

Nota: in caso due percorsi o sotto-percorsi richiedano lo stesso quantitativo di carburante, viene scelto quello che attraversa meno città. In caso di ulteriore parità, è selezionato quello che attraversa la città con ID maggiore (poiché si pensa che l'ID aumenti con avvicinandosi alle *Rovine Perdute*).

Nota: è possibile che una città *A* sia collegata ad una città *B* tramite un collegamento, ma che *non esista* il collegamento inverso. Pertanto, nel calcolo del percorso, si deve considerare che tale sentiero potrà essere attraversato **solo in un senso**.

Il veicolo *Tonatiuh*

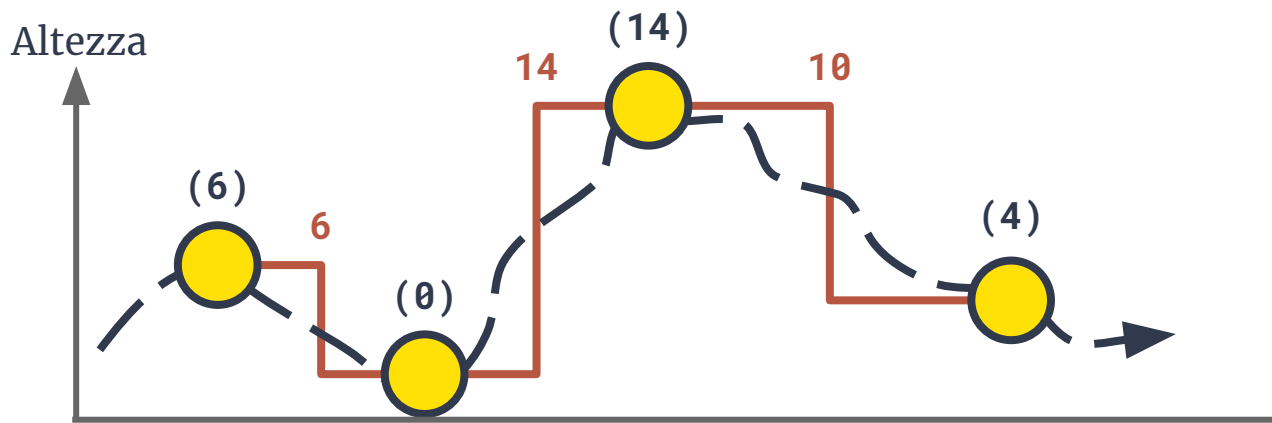
Il veicolo del *Team Tonatiuh* si muove fra due città consumando carburante in base alla **distanza euclidea** fra le due, calcolata prendendo le rispettive coordinate X e Y secondo la classica formula della “distanza fra due punti”. Il costo del carburante per il veicolo *Tonatiuh* non varia in funzione dell’altitudine delle città.



Distanza totale: $5 + 5 + 13 = 23$, pertanto verranno spese 23 unità di carburante.

Il veicolo *Metztli*

Il veicolo del *Team Metztli*, al contrario, spende unità di carburante in funzione della **differenza di altitudine** fra due città, sia esso un percorso **in salita o in discesa**; in pratica, la distanza fra le città è data dal modulo della differenza delle loro altitudini. La posizione (in coordinate cartesiane) delle città non influisce sul consumo del veicolo *Metztli*.



Distanza totale: $6 + 14 + 10 = 30$, pertanto verranno spese 30 unità di carburante.

Il documento finale

Una volta calcolato il percorso ottimo per ciascuno dei due team, non vi resta che riportare la soluzione in un **documento finale** (denominato “Routes.xml”) da consegnare agli esploratori. Il documento finale sarà così strutturato:

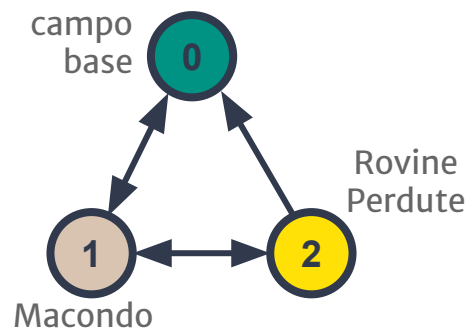
```
<routes>
  <route team="Tonatiuh" cost="carburante_speso" cities="num_città_toccate">
    <city id="" name=""> // Elenco delle città nel percorso
    <city id="" name="">
    <city ...>
    ...
  </route>
  <route team="Metztli" ...> // Stessa cosa per il secondo team
    <city ...>
    ...
  </route>
</routes>
```

Esempio 6.2.3

In continuazione all'**Esempio 6.2.2** riporto qui il relativo documento conclusivo contenente i due percorsi dal campo base alle *Rovine Perdute*:

```
<routes>
  <route team="Tonatiuh" cost="18" cities="3">
    <city id="0" name="campo base">
    <city id="1" name="Macondo">
    <city id="2" name="Rovine Perdute">
  </route>
  <route team="Metztli" cost="9" cities="3">
    <city id="0" name="campo base">
    <city id="1" name="Macondo">
    <city id="2" name="Rovine Perdute">
  </route>
</routes>
```

Schema dei collegamenti:



Nota: il campo base ha sempre id pari a zero, le *Rovine Perdute* hanno sempre id pari al numero di città meno 1.

Modalità di consegna

Una volta terminato il programma, esso dovrà essere caricato su piattaforma **GitHub** tramite l'account del responsabile del gruppo, all'interno di una **nuova** repository.

La repository dovrà chiamarsi “PgAr2019_NomeGruppo_LeRovinePerdute”, con il nome del vostro gruppo in *capitalized camel case* al posto della stringa “NomeGruppo”. Tale repository dovrà contenere l'intero progetto *Eclipse* (o di un altro IDE) del vostro gruppo con le eventuali dipendenze, in modo che poi si possa clonare e sia già funzionante.

L'intero progetto è da consegnare entro la data comunicata via mail. Eventuali modifiche successive all'orario di consegna non verranno valutate.

Note sulla valutazione

Oltre a valutare – come di consueto – il programma nel suo complesso, verranno valutate nello specifico le seguenti caratteristiche:

- Presenza di **documentazione**.
- Progettazione e implementazione dell'algoritmo per la **ricerca del percorso minimo**, nei due casi previsti.
- Implementazione di **lettura e scrittura** dei dati su file XML.
- **Correttezza** della soluzione fornita dal programma (il programma verrà testato con un software di testing automatico).
- **Solidità** del programma di fronte a scelte “peculiari” dell'utente (ossia: quanto è a prova di idiota il vostro codice? Avete previsto *tutti* i casi particolari?).
- Eventuali **scelte di implementazione** originali e interessanti che migliorino l'efficienza e l'esperienza dell'utente.
- Utilizzo corretto di **Git** e **GitHub**.

Bibliografia e risorse online

- Come probabilmente avrete intuito, il calcolo del percorso minimo è effettuabile mediante l'algoritmo di **Dijkstra**. Qui potete trovare lo pseudocodice relativo: https://it.wikipedia.org/wiki/Algoritmo_di_Dijkstra
- Leaked image of the *Lost Ruins*: click [here!](#) (Per entrare nel *mood* dell'esercizio).
- Soundtrack ufficiale dell'esercizio: www.youtube.com/VJxNv2m7qns.



Questa presentazione è stata realizzata da
Michele Dusi
ed è stata aggiornata da
Daniele Della Morte
per lo Student Branch IEEE
dell'Università degli Studi di Brescia,
in occasione del Programma Arnaldo 2019.

*Si prega di non modificare o distribuire il contenuto di tale documento
senza essere in possesso dei relativi permessi.*



UNIVERSITÀ
DEGLI STUDI
DI BRESCIA



IEEE