

## Secondo Progetto ASD 2016/17

# La vendetta del Re Lich

C'era una volta...



'era una volta, nel  
regno di *Algorìtmia*, un Re illuminato  
e benevolo di nome *Albertus I*.



## C'era una volta...

**S**'era una volta, nel regno di *Algorìtmia*, un Re illuminato e benevolo di nome *Albertus I*.

La vita scorreva tranquilla e prospera, con floridi commerci tra le varie città del regno. Ogni strada del regno poteva essere percorsa in un dato numero di giorni, noto per ciascuna di esse.



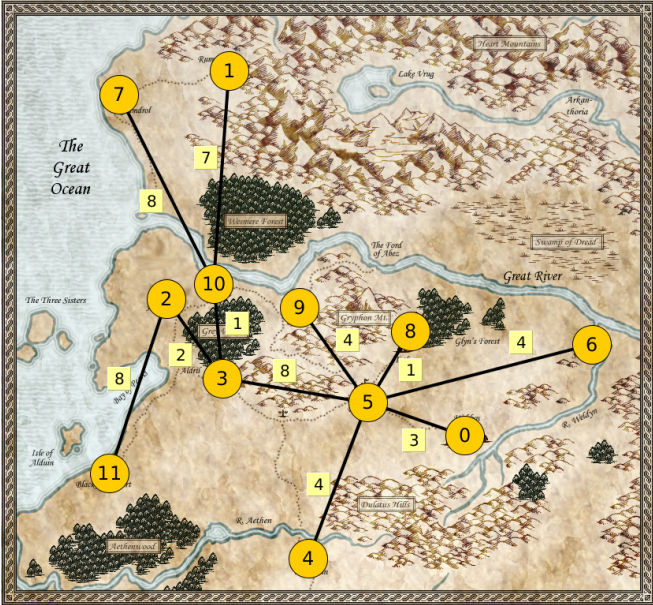
## C'era una volta...

**S**'era una volta, nel regno di *Algorìtmia*, un Re illuminato e benevolo di nome *Albertus I*.

La vita scorreva tranquilla e prospera, con floridi commerci tra le varie città del regno. Ogni strada del regno poteva essere percorsa in un dato numero di giorni, noto per ciascuna di esse. Per semplificare la logistica, il Re aveva decretato che tra due città ci fosse sempre un unico percorso.



# Mappa di Algoritmia



## Il segreto del Re

Pochissimi sapevano che Re Albertus avesse un fratello gemello, *Malbertus*.



## Il segreto del Re

Pochissimi sapevano che Re Albertus avesse un fratello gemello, *Malbertus*.

Al contrario del fratello, *Malbertus* aveva un cuore malvagio e invidioso. Fu allontanato dalla famiglia reale appena raggiunse la maggiore età.





## Il segreto del Re

Pochissimi sapevano che Re Albertus avesse un fratello gemello, *Malbertus*.

Al contrario del fratello, *Malbertus* aveva un cuore malvagio e invidioso. Fu allontanato dalla famiglia reale appena raggiunse la maggiore età. Pieno di rancore per essere stato cacciato, *Malbertus* giurò vendetta e iniziò a studiare le Arti Proibite. In segreto pianificava un attacco al Regno con un'armata di non-morti, con l'obiettivo di conquistare Algoritmia e prendere il posto di suo fratello sul trono.



# Il Nemico

Per avere ancora più potere,  
Malbertus decise di diventare un  
non-morto egli stesso.

## Il Nemico

Per avere ancora più potere, Malbertus decise di diventare un non-morto egli stesso. Con un ultimo oscuro incantesimo compì la trasformazione in un *Lich*.

## Il Nemico

Per avere ancora più potere, Malbertus decise di diventare un non-morto egli stesso. Con un ultimo oscuro incantesimo compì la trasformazione in un *Lich*. Da allora si fa chiamare...

## Il Nemico



Per avere ancora più potere, Malbertus decise di diventare un non-morto egli stesso. Con un ultimo oscuro incantesimo compì la trasformazione in un *Lich*. Da allora si fa chiamare...

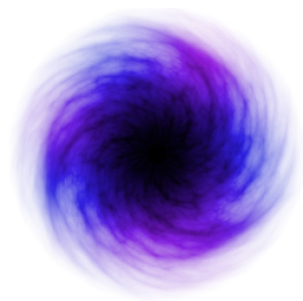
**Mal Mortèxor, il Re Lich**

# Un pericolo imminente

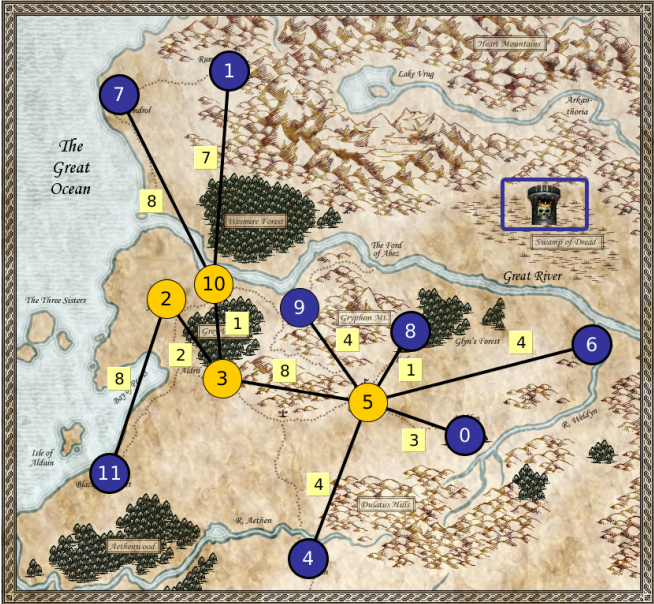
L'attacco di Mal Mortèxor punta a colpire inizialmente le città più deboli e isolate, quelle collegate ad una sola altra città vicina.

## Un pericolo incombente

L'attacco di Mal Mortèxor punta a colpire inizialmente le città più deboli e isolate, quelle collegate ad una sola altra città vicina. Con una magia, ha creato dei portali che collegano il suo antro con queste città. La sua armata avrebbe iniziato l'invasione attraversando questi portali.



# Posizionamento dei portali





# Una Nuova Speranza

Fortunatamente, due fidati esploratori del Re hanno scoperto questi portali e hanno suggerito di preparare un contrattacco sfruttando proprio quei collegamenti.



# Una Nuova Speranza

Fortunatamente, due fidati esploratori del Re hanno scoperto questi portali e hanno suggerito di preparare un contrattacco sfruttando proprio quei collegamenti. I portali permettono a chiunque li attraversi di essere teletrasportato immediatamente all'antro di Mal Mortèxor.



# Cavalieri Pavidì

Un attacco coordinato e in forze è l'unica possibilità per salvare Algoritmia! Il Re chiama a raccolta i suoi cavalieri e tutti coloro siano in grado di combattere.

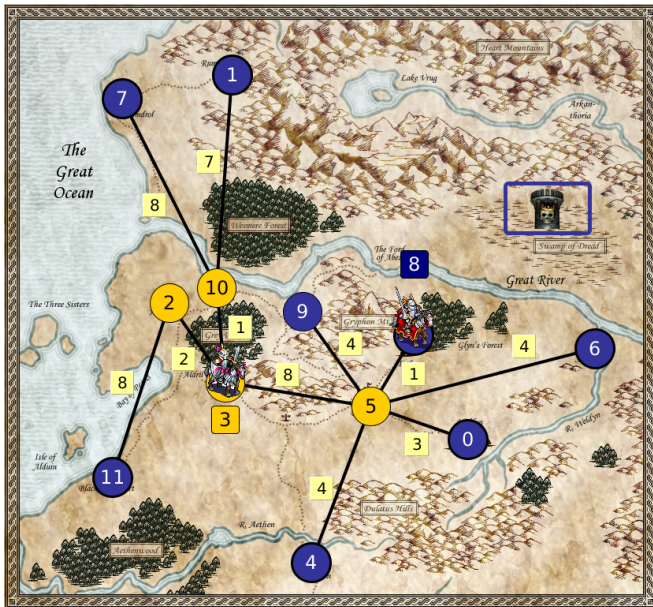


# Cavalieri Pavidì

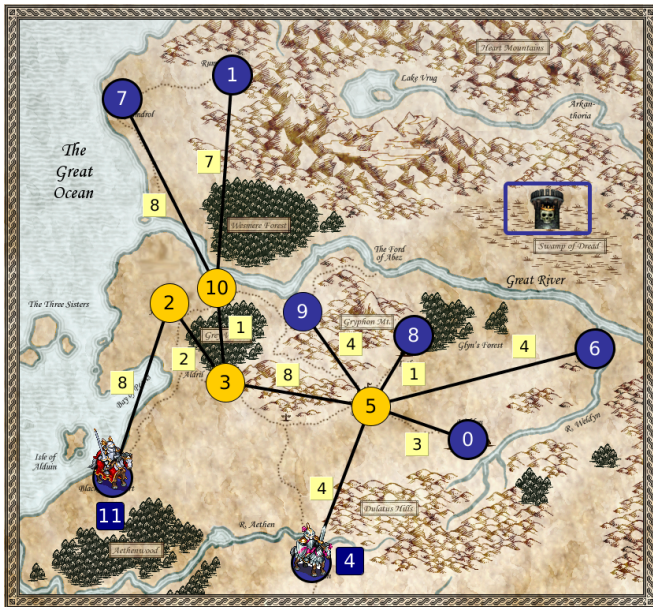
Un attacco coordinato e in forze è l'unica possibilità per salvare Algoritmia! Il Re chiama a raccolta i suoi cavalieri e tutti coloro siano in grado di combattere. Purtroppo, i cavalieri del Re hanno paura dei non-morti e quindi una volta ottenuto il via ad attaccare ognuno di loro sceglierà il portale **più lontano**.



# Posizioni di partenza ( $c_1 : 8, c_2 : 3$ )



# Portali più lontani ( $c_1 : 8 \rightarrow 11, c_2 : 3 \rightarrow 4$ )

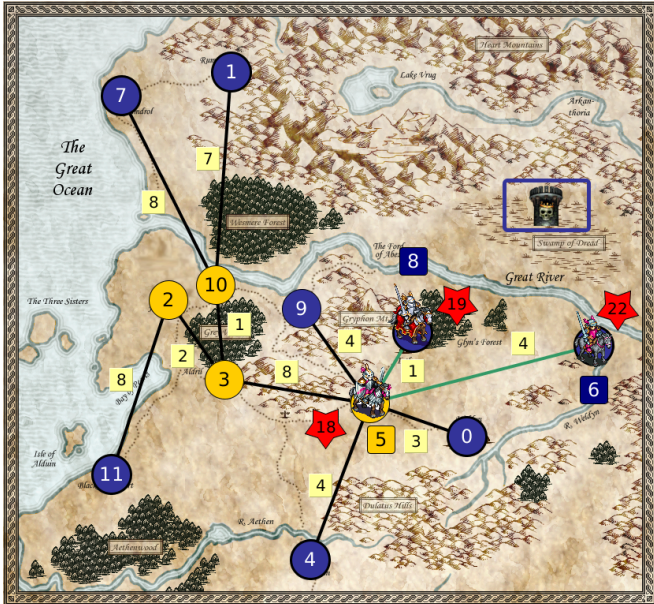


# Al Contrattacco!

Il Re è un ottimo stratega e conosce bene le sue truppe, quindi stabilisce questi requisiti:

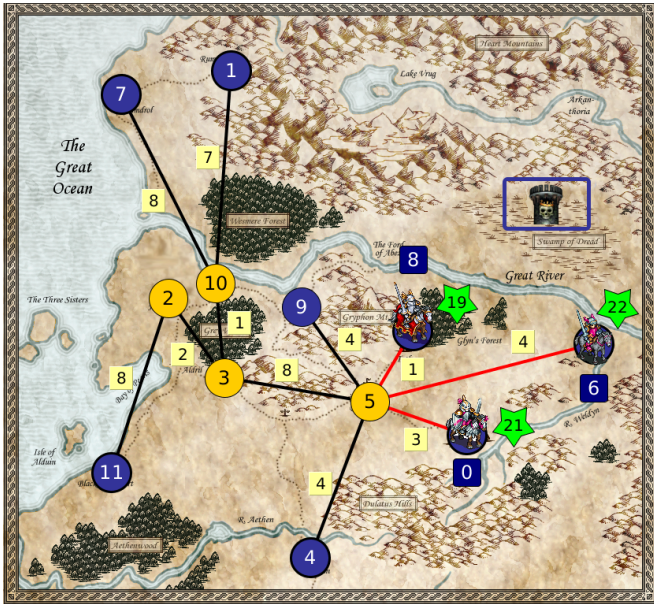
- ▶ L'attacco deve essere il più simultaneo possibile, ovvero l'intervallo fra l'arrivo del primo cavaliere e l'arrivo dell'ultimo deve essere al massimo  $L$  giorni;
- ▶ Per semplificare l'addestramento delle truppe, le posizioni iniziali dei cavalieri devono essere in città contigue.

$L = 3$ , soluzione non valida (intervallo di attacco  $> L$ )

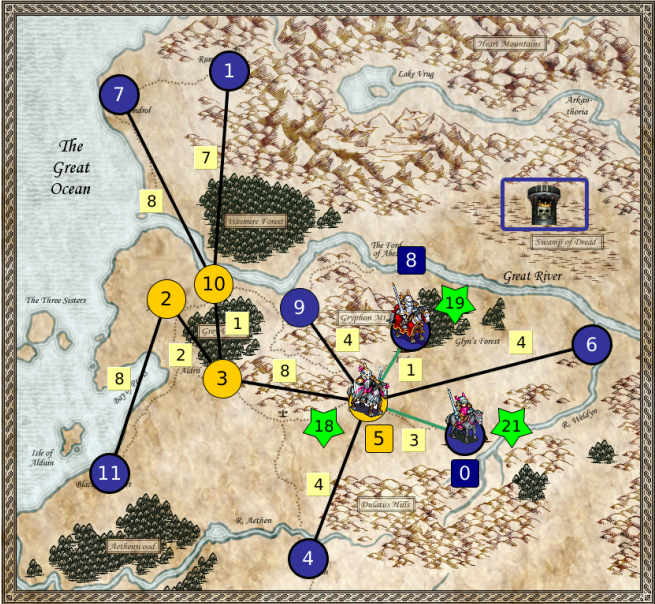




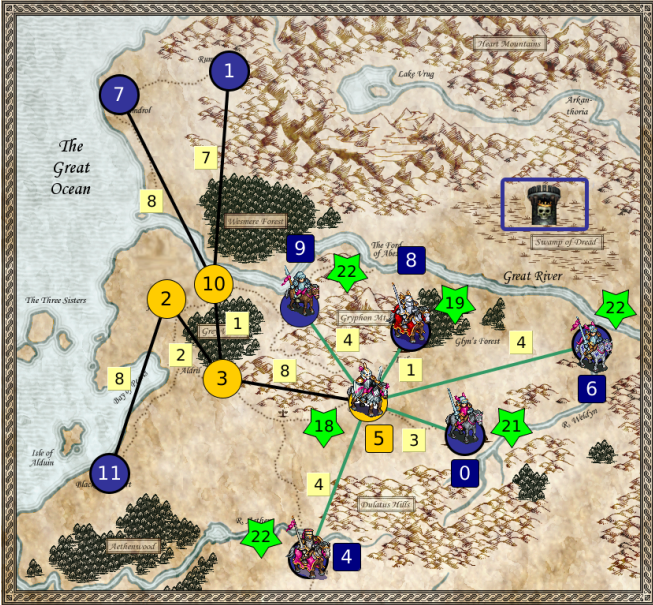
$L = 3$ , soluzione non valida (posizioni iniziali non contigue)



$L = 3$ , soluzione corretta: 3 cavalieri



$L = 4$ , soluzione corretta: 6 cavalieri



# Il Piano di Guerra

## Problema

Il vostro compito, come consiglieri di guerra del Re, è di calcolare il numero massimo  $C$  di cavalieri con cui sferrare l'attacco che rispetti i requisiti, per diversi valori di  $L$ .

# Input/Output (I)

## Input

Un file di testo con  $i_1 + i_2$ :

- $i_1$ : ( $N$  righe) La mappa di Algoritmia, ovvero l'albero pesato con le città, le strade che le collegano e i tempi di percorrenza:
- ▶ la prima riga contiene il numero  $N$  di città;
  - ▶ le successive  $N - 1$  righe contengono ciascuna 3 interi ( $p, a, w$ ): città di partenza, città di arrivo, peso (giorni di percorrenza);
- $i_2$ : (2 righe) La specifica dei valori di  $L$  per i quali dovete risolvere il problema:
- ▶ una riga con un intero  $K$  che indica quanti valori diversi di  $L$  dovete calcolare;
  - ▶ una riga con  $K$  interi separati da spazio  $L_1 L_2 \dots L_K$ , i vari valori di  $L$  per cui dovete risolvere il problema;

## Output

- o:  $K$  righe, il risultato richiesto  $C_{L_i}$  per ciascun valore  $L_i$  dato in input (in ordine);

# Input/Output

12	1
0 5 3	2
1 10 7	3
2 3 2	3
2 11 8	6
3 5 8	6
3 10 1	
4 5 4	
5 6 4	
5 8 1	
5 9 4	
7 10 8	
6	
0 1 2 3 4 5	

# Note

## Assunzioni:

- ▶ L'albero è non-orientato
- ▶ L'albero è connesso
- ▶  $1 \leq N \leq 400000$
- ▶  $1 \leq w \leq 1000$
- ▶  $1 \leq K \leq 10$
- ▶  $1 \leq L \leq 1000000$

Leggete bene formato di input ed output sul testo su judge

# Punteggio e casi di test

## Punteggio

1. soluzione corretta: 5 punti;
2. soluzione errata: 0 punti;

## Casi di test

- ▶ 20 casi di test totali;
- ▶ In 6 casi l'albero è una linea non pesata;
- ▶ In 10 casi l'albero è una linea;
- ▶ In 15 casi l'albero ha  $N \leq 10000$  nodi;

Per superare il progetto (e accedere all'esame)  $\geq 30$  punti



## Note varie

- ▶ Il progetto darà fino a 2 punti bonus allo scritto
- ▶ Conta il punteggio dell'ultimo sorgente accettato da judge
- ▶ Scadenza **mercoledì 21 Dicembre alle 20:00**
- ▶ Limite di 40 sottoposizioni per gruppo
- ▶ Potete provare con un dataset equivalente sulla vostra macchina  
(scaricate dal sito: [judge.science.unitn.it/slides/](http://judge.science.unitn.it/slides/))

# Do's

È permesso:

1. Discutere all'interno del gruppo
2. Chiedere chiarimenti sul testo
3. Chiedere opinioni su soluzioni
4. Sfruttare codice fornito nei laboratori
5. Utilizzare pseudocodice da libri o Wikipedia
6. Richiedere aiuto (anche pesante) per la soluzione "minima"
7. Venire a ricevimento

# Dont's

È vietato:

1. Discutere con altri gruppi
2. Mettere il proprio codice su repository pubblici
3. Utilizzare codice scritto da altri
4. Condividere codice (!!!!!!!!!!!!!)

# Ricevimento

Nei seguenti giorni e orari mi trovate nell'aula indicata:

- ▶ Mercoledì 14/12/2016
  - ▶ 14:00 - 16:00 — aula Molveno (Povo 1)
- ▶ Lunedì 19/12/2016
  - ▶ 09:00 - 11:00 — Lases (Povo 2)
  - ▶ 14:00 - 16:00 — Lases (Povo 2)
- ▶ Venerdì 16/12/2016
  - ▶ **Laboratorio: 14:00 - 16:00 — B106 (Povo 2)**
- ▶ Martedì 20/12/2016
  - ▶ 09:00 - 13:00 — Levico (Povo 2)
- ▶ Mercoledì 21/12/2016
  - ▶ **Laboratorio: 11:00 - 13:00 — B106 (Povo 2)**
  - ▶ 14:00 - 16:00 — Ofek (Povo 1)

⇒ scrivetemi via mail ([cristian.consonni@unitn.it](mailto:cristian.consonni@unitn.it)) prima di venire, per favore

⇒ <http://bit.ly/ASDLabRicevimento>

# Ricevimento

GMT+01	mer 14/12	gio 15/12	ven 16/12	sab 17/12	dom 18/12	lun 19/12	mar 20/12	mer 21/12
07:00								
08:00								
09:00						09:00 – 11:00 Ricevimento Progetto 2 - Laboratorio ASD aula Lases (Povo 2)	09:00 – 13:00 Ricevimento Progetto 2 - Laboratorio ASD aula Levico (Povo 2)	
10:00								
11:00								11:00 – 13:00 Laboratorio ASD aula B106 (Povo 2)
12:00								
13:00								
14:00	14:00 – 16:00 Ricevimento Progetto 2 - Laboratorio ASD aula Molveno (Povo 1)		14:00 – 16:00 Laboratorio ASD aula B106 (Povo 2)			14:00 – 16:00 Ricevimento Progetto 2 - Laboratorio ASD aula Lases (Povo 2)		14:00 – 16:00 Ricevimento Progetto 2 - Laboratorio ASD aula Ofak (Povo 1)
15:00								
16:00								
17:00								
18:00								

# Credits

Tutte le immagini e le icone provengono dal gioco "*The Battle for Wesnoth*" rilasciato con licenza GNU GPLv3.  
(<https://www.wesnoth.org/>).

La licenza GNU GPL è una licenza libera, per maggiori informazioni e una copia del testo della licenza si veda:  
<http://www.gnu.org/licenses/>.