

## Algoritmi e Strutture Dati

# La vendetta del Re Lich (lich)

### Testo del problema

Slides originali su: [judge.science.unitn.it/slides/asd16/prog2.pdf](http://judge.science.unitn.it/slides/asd16/prog2.pdf)

C'era una volta, nel regno di *Algoritmia*, un Re illuminato e benevolo di nome *Albertus I*. La vita scorreva felice nel regno ed il Re era molto benvoluto dai suoi sudditi. Gli abitanti di Algoritmia si rivolgevano sempre a lui per ogni loro problema. Il Re era molto saggio – avendo studiato da bambino sotto la guida del maestro *al-Khwarizmi* – e, dopo aver ascoltato pazientemente ogni questione, proproneva sempre ai suoi sudditi soluzioni efficaci ed efficienti per i loro problemi.

Sotto la guida di Albertus il regno prosperava, con floridi commerci tra le varie città del regno, collegate da strade ben mantenute e sorvegliate. Ogni strada del regno poteva essere regolarmente percorsa in un numero prefissato di giorni, noto per ciascuna di esse. Per semplificare la logistica e l'organizzazione del regno, il Re aveva decretato che tra due città ci fosse sempre un unico percorso. La Figura 1 mostra una mappa di Algoritmia.

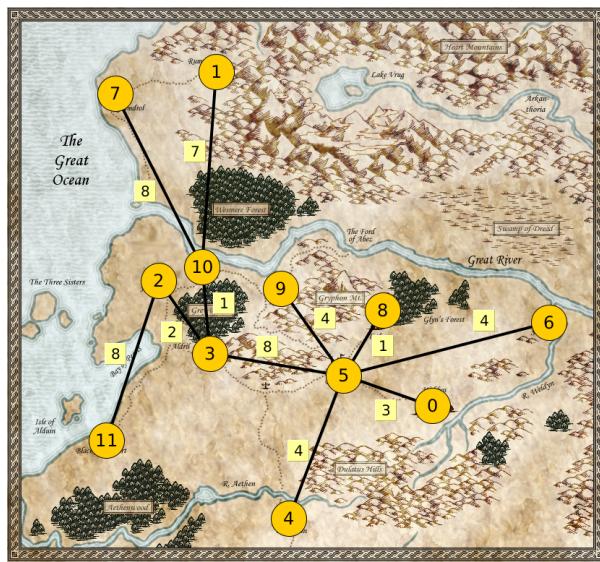


Figura 1: Mappa delle città e delle strade di Algoritmia.

Purtroppo, la felicità e la pace di Algoritmia non erano destinate a durare: un Nemico tramava per distruggerle. Un Nemico nato nella stessa casa del Re. Pochi ricordano che Re Albertus avesse un fratello gemello, *Malbertus*. *Malbertus* era del tutto identico ad Albertus, tranne che nel cuore: al contrario del fratello era malvagio e invidioso. Per questo, fu allontanato dalla famiglia reale appena raggiunse la maggiore età e tutti ben presto si dimenticarono di lui.

Pieno di rancore per essere stato cacciato, il fratello dimenticato giurò vendetta e inizio a studiare le Arti Proibite e - motivato dalla sua sinistra coscienza - diventò un potente necromante. La sua sete di vendetta non aveva confini: il suo obiettivo era conquistare Algoritmia e prendere il posto di suo fratello sul trono.

Con i suoi poteri di necromante, da anni pianificava un attacco al Regno con un'armata di non-morti al suo comando. Per avere ancora *Malbertus* decise di diventare un non-morto egli stesso. Con un ultimo oscuro incantesimo compì la trasformazione e diventò il più potente non-morto, un Lich. Da allora si fa chiamare **Mal Mortèxor, il Re Lich**.

Con il suo potere al massimo, dal suo antro nella Palude del Terrore, una zona disabitata a nord-ovest di Algoritmia, aveva generato decine di scheletri-soldato pronti ad attaccare. Il suo piano d'attacco puntava a colpire inizialmente le città più deboli e isolate, quelle collegate ad una sola altra città vicina. Con una magia, aveva creato dei portali che collegano il suo antro con queste città. La sua armata avrebbe iniziato l'invasione attraversando questi portali. La Figura 2 mostra la disposizione dei portali creati da Mal Mortèxor.

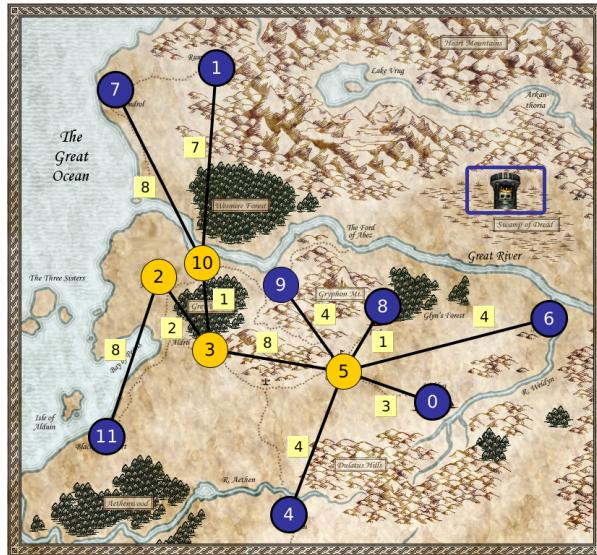


Figura 2: La disposizione dei portali creati da Mal Mortèxor.

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

Informato della situazione, il Re pensa subito a una soluzione: 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**. Nelle Figure 3 e 4 è mostrato un esempio per due cavalieri.

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.

Ovviamente, quante più forze attaccheranno Mal Mortèxor tanto più alte saranno le possibilità di vittoria, quindi il Re vuole massimizzare il numero di cavalieri che parteciperanno al contrattacco.

Venite convocati quali membri del *Concilio di guerra* di Algoritmia e vi vengono presentati i requisiti richiesti dal Re: serve il vostro aiuto per disporre i cavalieri. Il Re vuole esplorare svariate possibilità per durate dell'attacco diverse (valori di  $L$  diversi). Il vostro compito, come consiglieri di guerra del Re, è di calcolare il numero massimo di armate con cui sferrare l'attacco per diversi valori di  $L$  e rispettando i requisiti.

Le Figure 5 e 6 mostrano una soluzione per i casi  $L = 3$  e  $L = 4$  rispettivamente.

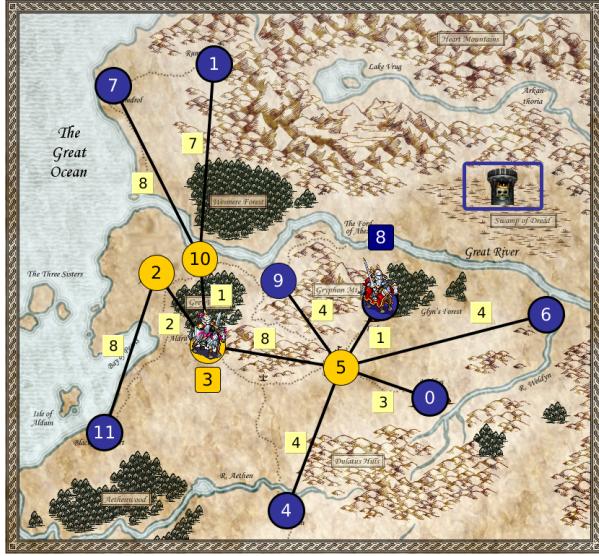


Figura 3: Due cavalieri  $c_1$  e  $c_2$  che partono rispettivamente dalle posizioni 8 e 3.

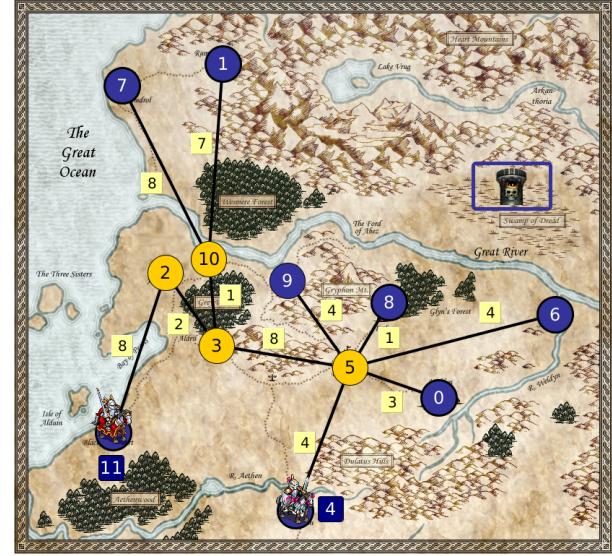


Figura 4: Le posizioni di arrivo dei due cavalieri, rispettivamente  $c_1 : 8 \rightarrow 11, c_2 : 3 \rightarrow 4$ .

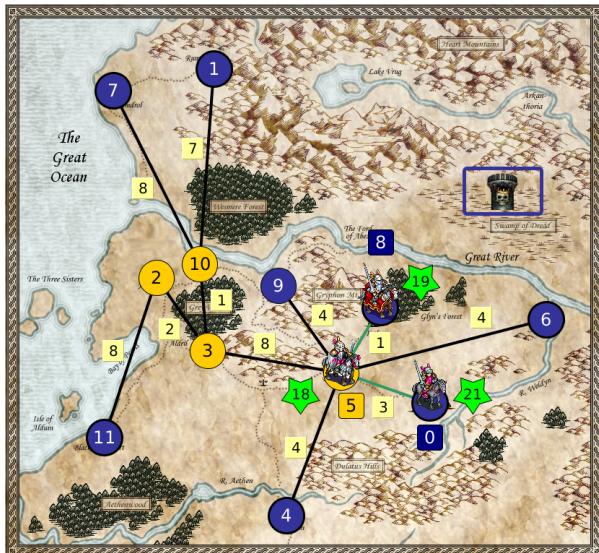


Figura 5: Soluzione corretta per il caso  $L = 3$ : 3 cavalieri.

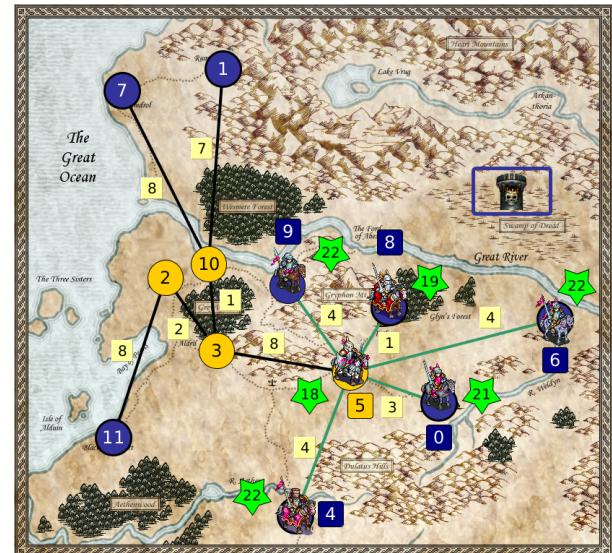


Figura 6: Soluzione corretta per il caso  $L = 4$ : 6 cavalieri.

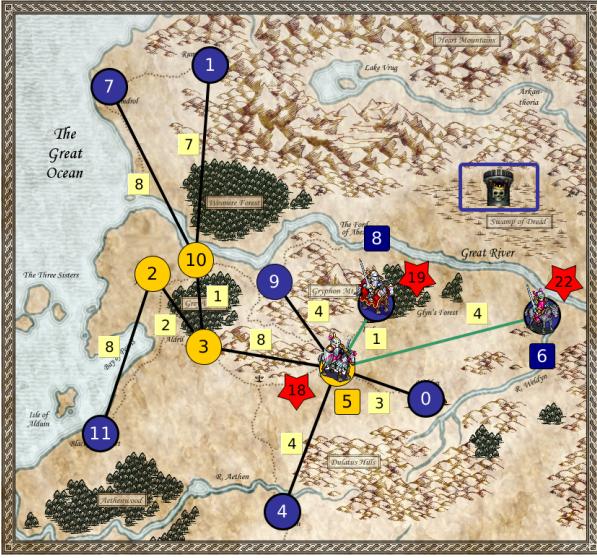


Figura 7: Con  $L = 3$  questa disposizione non rispetta il primo requisito: la differenza tra l'arrivo del primo e dell'ultimo cavaliere è  $4 > 3 = L$ .

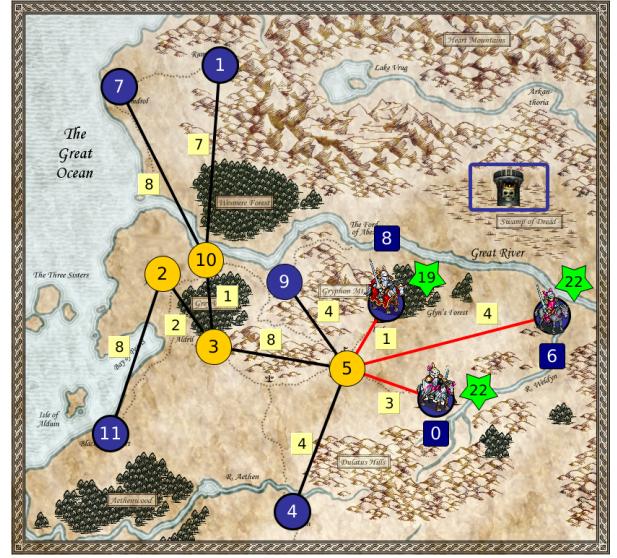


Figura 8: Con  $L = 3$  questa disposizione non rispetta il secondo requisito: le posizioni iniziali dei cavalieri non occupano città contigue.

Le Figure 7 e 8 presentano soluzioni non valide per il caso  $L = 3$ .

- La disposizione in Figura 7 non rispetta il primo requisito: la differenza tra l'arrivo del primo cavaliere (quello che parte dalla città 5) e l'arrivo dell'ultimo (quello che parte dalla città 6) è maggiore di  $L$ ;
- La disposizione in Figura 8 non rispetta il secondo requisito: le posizioni iniziali dei cavalieri non occupano città contigue.

Vi viene fornita la completa del regno, ovvero l'albero delle città e delle strade di Algoritmia. Inoltre vi vengono indicati i valori delle durata totali dell'attacco ( $L$ ) che dovete considerare. Dovrete indicare qual è il numero massimo  $C_{L_i}$  di cavalieri che potete posizionare per ciascun valore di  $L_i$  dato (in ordine).

## Formato dell'input

L'input è costituito da un file di testo di  $N + 2$  righe costituito da  $i_1 + i_2$ :

$i_1$ : ( $N$  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;

## **Formato dell'output**

Un file contenente  $K$  righe, il numero massimo di cavalieri che si possono posizionare sulla mappa rispettando i requisiti per ciascun valore di  $L$  dato in input (in ordine);

## **Assunzioni**

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

## **Note**

Si accede all'esame totalizzando almeno 30 punti. Vengono utilizzati 20 casi di test che possono dare al massimo 5 punti ciascuno. Il punteggio massimo è di 100 punti.

## **Punteggio**

Per ogni caso di test, le soluzioni vengono valutate nel modo seguente:

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

## **Descrizione dei casi di test**

- 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;

## Esempi di input/output

File input.txt	File output.txt
12 0 5 3 1 10 7 2 3 2 2 11 8 3 5 8 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	1 2 3 3 6 6
File input.txt	File output.txt
11 0 5 1 1 4 1 1 6 1 2 3 1 2 8 1 3 7 1 4 5 1 6 10 1 7 9 1 8 10 1 6 0 1 2 3 4 5	1 3 5 7 9 11