

TEMPO A DISPOSIZIONE: 1 H 30 MIN

1. Considerate il problema descritto nel seguito. Trasformate l'input in un grafo e applicate uno degli algoritmi visti a lezione per risolverlo. Fornite le seguenti informazioni:

- Quali sono i vertici? Cosa rappresenta ciascun vertice?
- Quali sono gli archi? Sono orientati o non orientati?
- Se i vertici e/o gli archi hanno dei valori associati, quali sono questi valori?
- Quale problema si deve risolvere sul grafo?
- Quale algoritmo si sta usando per risolvere il problema?
- Qual è il tempo di esecuzione dell'intero algoritmo, *incluso* il tempo di creazione del grafo, *in funzione dei parametri di input originali*?

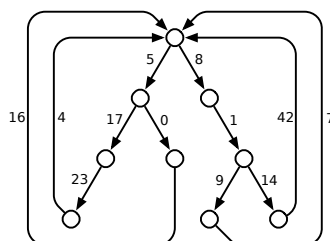
Scale e Serpenti è un gioco da tavolo tradizionale, nato in India e diffuso soprattutto nei paesi di lingua inglese (il nome originale è “snakes and ladders”). Si gioca su una scacchiera quadrata di dimensione $n \times n$ con le celle numerate consecutivamente da 1 a n^2 , iniziando dalla cella in basso a sinistra e proseguendo riga per riga dal basso verso l'alto, alternando la numerazione delle righe verso sinistra e verso destra. Alcune coppie di celle, poste su righe diverse, sono collegate da “serpenti” che portano verso il basso, o da “scale” che portano verso l'alto, come nell'esempio sottostante. *Ogni casella può essere l'estremo di al massimo un serpente o una scala.*

| | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|
| 100 | 99 | 98 | 97 | 96 | 95 | 94 | 93 | 92 | 91 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 80 | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Il gioco inizia posizionando una pedina nella cella 1. Ad ogni mossa, la pedina può avanzare di un numero di celle fino ad un massimo di k (tipicamente 6). Se al termine della mossa la pedina si trova nella cella *superiore* di un serpente, allora è *obbligata* a scivolare giù lungo il serpente fino alla cella inferiore. Se invece si trova nella cella *inferiore* di una scala, allora *può* arrampicarsi verso l'alto fino a raggiungere la cella superiore.

Descrivere ed analizzare un algoritmo efficiente per calcolare il numero minimo di mosse necessarie per raggiungere la cella finale.

2. Un *albero ad anello* è un grafo orientato e pesato costruito aggiungendo ad ogni foglia di un albero binario un arco che porta alla radice. Tutti gli archi hanno peso non negativo.



Un esempio di albero ad anello.

- Quanto tempo impiega l'algoritmo di Dijkstra per calcolare il percorso più breve da un vertice u ad un vertice v in un albero ad anello con n nodi?
- Descrivere e analizzare un algoritmo più veloce.

3. Dimostrare che un problema X è NP-hard richiede diversi passaggi:

- Scegli un problema Y che sai essere NP-hard (perché l'hai visto a lezione).
- Descrivi un algoritmo per risolvere Y usando un algoritmo per X come subroutine. Tipicamente questo algoritmo ha la seguente forma: data un'istanza di Y , trasformala in un'istanza di X , quindi chiama l'algoritmo magico black-box per X .
- Dimostra che l'algoritmo è corretto. Ciò richiede sempre due passaggi separati, che di solito hanno la seguente forma:
 - Dimostra che il tuo algoritmo trasforma istanze “buone” di Y in istanze “buone” di X .
 - Dimostra che il tuo algoritmo trasforma istanze “cattive” di Y in istanze “cattive” di X .
Equivalentemente: Dimostra che se la tua trasformazione produce un'istanza “buona” di X , allora era partita da un'istanza “buona” di Y .
- Mostra che il tuo algoritmo per Y funziona in tempo polinomiale, a meno della chiamata (o delle chiamate) all'algoritmo magico black-box per X . (Questo di solito è banale.)

Un circuito Hamiltoniano in un grafo G è un ciclo che attraversa ogni vertice di G esattamente una volta. Stabilire se un grafo arbitrario contiene un circuito Hamiltoniano è un problema NP-hard.

Un *circuito toniano* in un grafo G è un ciclo che attraversa almeno *la metà dei vertici* di G esattamente una volta. Dimostrare che stabilire se un grafo contiene un circuito toniano è un problema NP-hard.