

Algorytmy i Struktury Danych
Egzamin/Zaliczenie 3 (14.IX.2023)
Zadanie A: Przykłady

Szablon rozwiązania:	egz3a.py
Złożoność akceptowalna (2pkt):	$O(n^3)$
Złożoność lepsza (+1pkt):	$O(n^2 \log n)$
Złożoność wzorcowa (+1pkt):	$O(n^2)$
Gdzie n to liczba zamków.	

Dobrycerz (czyli rycerz, który zawsze uprzejmie mówi “dzień dobry”) chce się przedostać z zamku s do zamku t . Mapa zamków dana jest w postaci grafu nieskierowanego G , gdzie każda krawędź ma wagę oznaczającą ile godzin potrzeba, żeby ją przebyć. Wagi to liczby naturalne ze zbioru $\{1, 2, \dots, 8\}$. Po najdalej 16 godzinach podróży Dobrycerz musi nocować w zamku. Warunki uprzejmości wymagają, żeby spędził w takim zamku 8 godzin (przejazd przez zamki, w których nie nocuje nie kosztuje dodatkowego czasu; szybko mówi “dzień dobry” strażnikom i jedzie dalej). Mapa z której korzysta Dobrycerz ma to do siebie, że liczba dróg jest proporcjonalna do liczby zamków. Czyli jeśli zamków jest n , to wiadomo, że dróg jest $O(n)$.

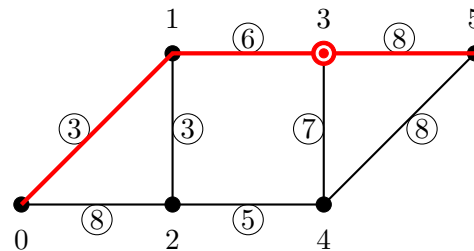
Zadanie polega na implementacji funkcji:

`goodknight(G, s, t)`

która na wejściu otrzymuje graf opisujący mapę zamków, reprezentowany w postaci macierzy sąsiedztwa (czyli $G[i][j]$ to liczba godzin, konieczna do przejechania bezpośrednio z zamku i do zamku j ; w przypadku braku drogi $G[i][j] = -1$), zamek startowy s oraz zamek docelowy t , i zwraca minimalny czas (wyrażony w godzinach) potrzebny na przejazd z s do t (Dobrycerz nigdy nie musi nocować ani w zamku s ani w zamku t). Można założyć, że zawsze istnieje trasa z zamku s do t .

Przykład 1. Dla wejścia:

```
#      0  1  2  3  4  5
G = [ [ -1, 3, 8,-1,-1,-1 ], # 0
      [ 3,-1, 3, 6,-1,-1 ], # 1
      [ 8, 3,-1,-1, 5,-1 ], # 2
      [ -1, 6,-1,-1, 7, 8 ], # 3
      [ -1,-1, 5, 7,-1, 8 ], # 4
      [ -1,-1,-1, 8, 8,-1 ] ] # 5
s = 0
t = 5
```



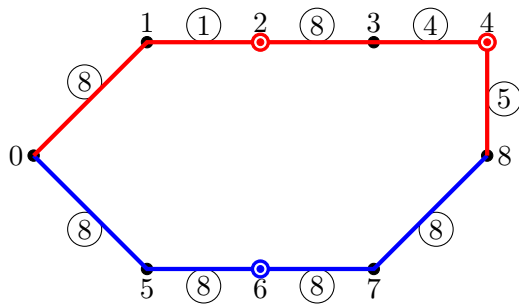
wynikiem jest 25. Dobrycerz pokonuje następującą trasę:

1. Jedzie z zamku 0 do zamku 1 (3 godziny).
2. Jedzie z zamku 1 do zamku 3 (6 godzin).
3. Nocuje w zamku 3 (8 godzin).
4. Jedzie z zamku 3 do zamku 5 (8 godzin)

Przykład 2 (najkrótsza ścieżka jest błędna). Rozważmy wejście:

```
#      0  1  2  3  4  5  6  7  8
G = [ [-1,  8, -1, -1, -1,  8, -1, -1, -1]    # 0
      [  8, -1,  1, -1, -1, -1, -1, -1, -1]    # 1
      [-1,  1, -1,  8, -1, -1, -1, -1, -1]    # 2
      [-1, -1,  8, -1,  4, -1, -1, -1, -1]    # 3
      [-1, -1, -1,  4, -1, -1, -1, -1,  5]    # 4
      [  8, -1, -1, -1, -1, -1,  8, -1, -1]    # 5
      [-1, -1, -1, -1, -1,  8, -1,  8, -1]    # 6
      [-1, -1, -1, -1, -1, -1,  8, -1,  8]    # 7
      [-1, -1, -1, -1,  5, -1, -1,  8, -1] ]  # 8

s = 0
t = 8
```



Poprawnym wynikiem jest niebieska ścieżka (0,5,6,7,8) o koszcie $8 + 8 + 8$ (nocleg) $+ 8 + 8 = 40$.

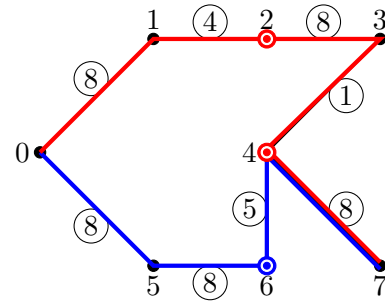
Czerwona ścieżka jest najkrótsza, jeśli pomija się noclegi (długość 26, podczas gdy niebieska ma długość 32). Ale uwzględniając noclegi jej koszt nie jest optymalny: $8 + 1 + 8$ (nocleg) $+ 8 + 4 + 8$ (nocleg) $+ 5 = 42$.

Przykład 3 (Zmodyfikowany Dijkstra). Część studentów stosowała modyfikację algorytmu Dijkstry. Jednym z typowych błędów takich modyfikacji było przechowanie dla danego zamku jedynie jego odległości od zamku początkowego, nie uwzględniając ile godzin rycerz jeszcze może podróżować. Ta informacja na ogół była wykorzystywana przy relaksacji, ale to nie wystarcza. Problem polega na tym, że dobrycerz może mieć kilka dróg prowadzących do danego zamku, dających różne pozostałe godziny (co gorsza, te drogi mogą mieć równą długość). Rozważmy następujący przykład:

```
#      0   1   2   3   4   5   6   7
G = [ [-1,  8, -1, -1, -1,  8, -1, -1] # 0
      [ 8, -1,  1, -1, -1, -1, -1, -1] # 1
      [-1,  1, -1,  8, -1, -1, -1, -1] # 2
      [-1, -1,  8, -1,  1, -1, -1, -1] # 3
      [-1, -1, -1,  1, -1, -1,  5,  8] # 4
      [ 8, -1, -1, -1, -1, -1,  8, -1] # 5
      [-1, -1, -1, -1,  5,  8, -1, -1] # 6
      [-1, -1, -1, -1,  8, -1, -1, -1] ] # 7
```

s = 0

t = 7



Poprawnym wynikiem jest niebieska ścieżka (0,5,6,4,7) o koszcie $8 + 8 + 8$ (nocleg) $+ 5 + 8 = 37$.

Błędna modyfikacja algorytmu Dijkstry może jednak wskazać ścieżkę czerwoną (0,1,2,3,4,7) o koszcie $8 + 4 + 8$ (nocleg) $+ 8 + 1 + 8$ (nocleg) $+ 8 = 45$. Problem polega na tym, że do zamku 4 prowadzą dwie ścieżki o równej długości, ale dające różne liczby godzin, przez które dobrycerz może jeszcze podróżować.

Wiele błędnych modyfikacji algorytmu Dijkstry prowadziło także do wyników 34 oraz 42, które zdają się nie odpowiadać naturalnym ścieżkom w grafie.