Grafy Eulerowskie

Marszruta przechodząca przez wszystkie krawędzie dokładnie raz - droga Eulera.
Zamknięta marszruta przechodząca przez wszystkie krawędzie dokładnie raz - cykl Eulera.

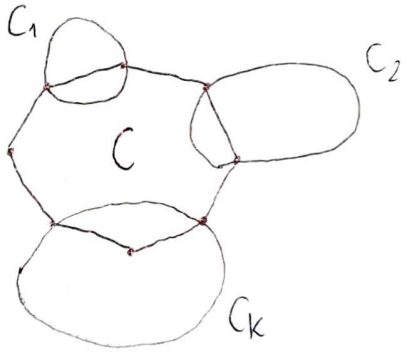
CYKL EULERA $\Leftrightarrow$ GRAF EULEROWSKI
DROGA EULERA $\Leftrightarrow$ GRAF PÓŁEULEROWSKI

Twierdzenie:

Graf G jest eulerowski wtw, gdy wszystkie wierzchołki mają stopień parzysty i wszystkie krawędzie są w jednej składowej spójnej.

$\Rightarrow$

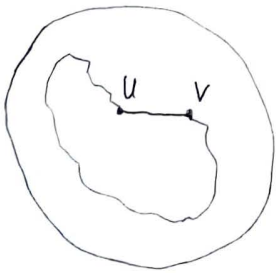
$\Leftarrow$ Indukcja po $m(G)$
W G musi istnieć cykl C



Twierdzenie:

Graf G jest półeulerowski wtw, gdy co najwyżej 2 wierzchołki mają stopień nieparzysty i wszystkie składowe są w jednej składowej spójnej.

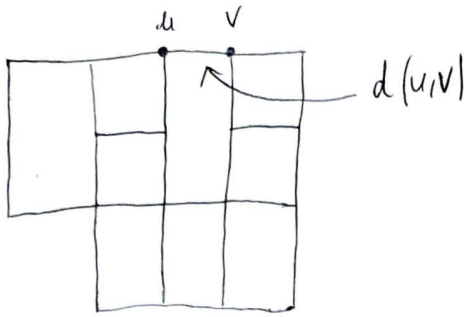
$\Leftarrow$



$\Rightarrow$ 1° 0 wierzchołków ma stopień nieparzysty $\Rightarrow$ G jest eulerowski

2° 2 wierzchołki (u,v) mają stopień nieparzysty:
Dodajemy krawędź $\{u,v\}$. W powstałym grafie jest cykl Eulera, który po wyjęciu $\{u,v\}$ zmienia się w drogę Eulera.

Problem Chińskiego listonosza



W danym grafie znaleźć ścieżkę, która przechodzi co najmniej jeden raz przez wszystkie krawędzie i wraca do wierzchołka startowego.
(więcej tutaj: https://eduinf.waw.pl/inf/alg/001_search/0139.php)

Drogi Hamiltona w grafie

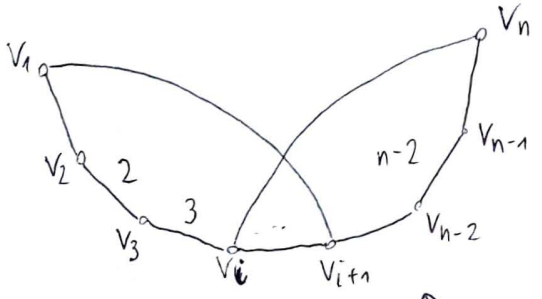
Droga Hamiltona to droga w G przechodząca dokładnie raz przez każdy wierzchołek, cykl Hamiltona to cykl w G...
Graf jest hamiltonowski jeśli posiada cykl Hamiltona, graf jest półhamiltonowski jeśli posiada drogę Hamiltona. Nie istnieje algorytm wielomianowy dla problemu czy G jest grafem hamiltonowskim lub półhamiltonowskim.

Twierdzenie (Ore)

Jeżeli G jest prosty, $n(G) \geq 3$ i $\forall u,v \in V(G): \{u,v\} \notin E(G)$
$\deg(u) + \deg(v) \geq n$,

to graf G zawiera cykl Hamiltona.

Dowód: Załóżmy, że G jest maksymalnym grafem spełniającym założenie twierdzenia i niezawierającym cyklu Hamiltona. G ma drogę Hamiltona.

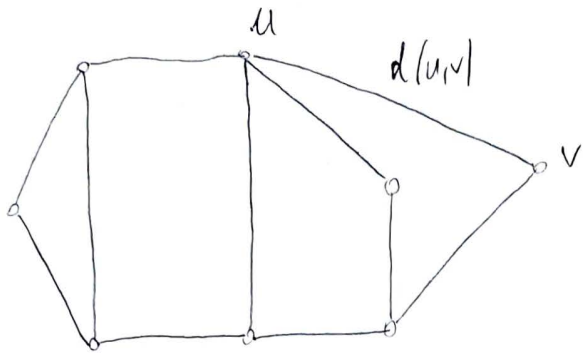


↑ cykl Hamiltona w G

SPRZECZNOŚĆ

$n-3$ - szufladki: $2, 3, \ldots, n-2$
Do szufladki i wpadają (jeśli istnieją) krawędzie $\{v_n, v_i\}$, $\{v_1, v_{i+1}\}$. Wtedy liczba krawędzi w szufladkach to:
$\deg(v_1) + \deg(v_n) - 2 \geq n-2$
Z zasady szufladkowej 2 krawędzie wpadną do 1 szufladki.

Problem komiwojażera (TSP)

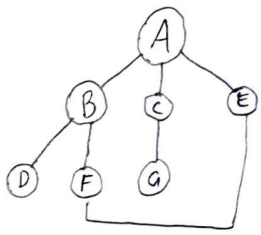


Dane jest n miast, które komiwojażer ma odwiedzić oraz odległości między nimi. Celem jest znalezienie najkrótszej drogi łączącej wszystkie miasta.

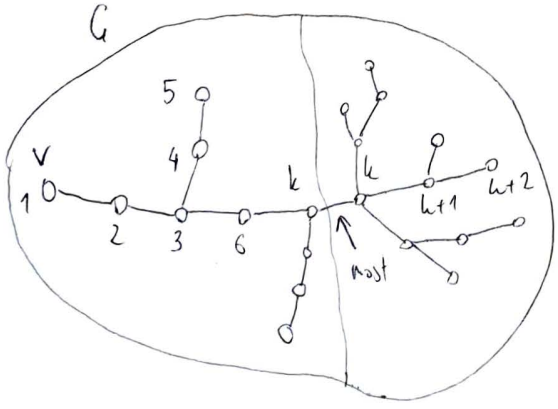
Przeszukiwanie grafów:

1° DFS - Depth First Search - przeszukiwanie w głąb (czas $O(n)$)

DFS (v):
zaznacz v jako odwiedzony (i wypisz)
Dla wszystkich $u \in N(v)$ nieodwiedzonych:
DFS(u)



DFS(A) → A, B, D, F, E, C, G



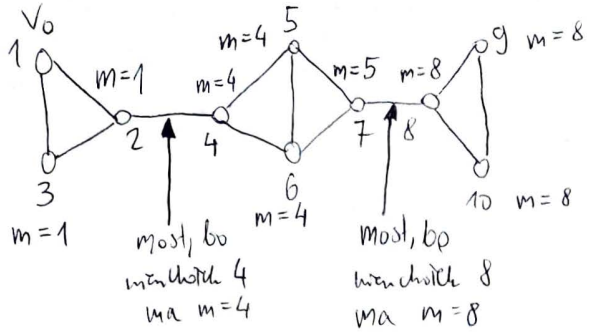
W m(v) będziemy liczyć min numer sąsiada niedrzewa z drzewa, którego korzeniem jest v (z wyłączeniem niedrzewa z którego dochodzimy do v).

$$m(v) = \min_{i,j} \left(m(u_i), nr(u_j) \right)$$

u_i - sąsiedzi v należący do drzewa o korzeniu w v

u_j - sąsiedzi v nienależący do tego drzewa i różni od niedrzewa z którego dochodzimy do v

$\{v, \text{poprzednik}(v)\}$ jest mostem $\Leftrightarrow$ $m(v) \geq k = nr(v)$



2° BFS - Breadth-First Search - przeszukiwanie wszerz (czas $O(m)$)

BFS(v):

kolejka $\leftarrow \{v_0\}$

v_0 zaznaczamy jako odwiedzony

Dopóki kolejka $\neq \emptyset$:

$v \leftarrow$ kolejka

Dla wszystkich sąsiadów $u \in N(v)$ nieodwiedzonych:

kolejka $\leftarrow u$

u zaznaczamy jako odwiedzony

Problem najkrótszych dróg

G - digraf, $d(u,v)$ - długość krawędzi $\{u,v\}$

1° Znajdowanie najkrótszej drogi z u do v

2° Znajdowanie najkrótszej drogi z v do wszystkich innych wierzchołków

3° Znajdowanie najkrótszej drogi między wszystkimi parami wierzchołków

Algorytm Dijkstry znajduje najkrótsze drogi z v_0 do wszystkich innych wierzchołków jeśli $d(u,v) > 0$ dla wszystkich krawędzi (u,v).

Alg Dijkstry (v_0):

$d[v_0] \leftarrow 0$, $V' \leftarrow \{v_0\}$

Dla $v \neq v_0$ jeśli $(v_0, v) \in E$: $d[v] \leftarrow d(v_0, v)$, $p[v] \leftarrow v_0$

wp.p.: $d[v] \leftarrow \infty$

Dopóki $V(G) \neq V'$:

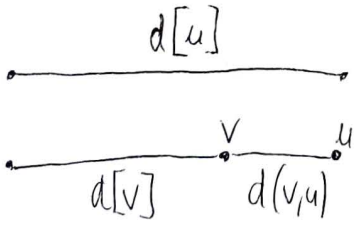
Wybierz z $V(G) \setminus V'$ wierzchołek v o minimalnym $d[v]$

$V' \leftarrow V' \cup \{v\}$

Dla $u \in N(v)$:

Jeśli $d[u] > d[v] + d(v,u)$ to $d[u] \leftarrow d[v] + d(u,v)$, $p[u] \leftarrow v$

Po każdej iteracji pętli „Dopóki" mamy spełnione dwa niezmienniki pętli, tzn. odległości do wierzchołków $v \in V'$ są poprawnie wyznaczone jako wartości $d[v]$ oraz w $d[v]$ dla $v \notin V'$ są wyznaczone długości najkrótszych dróg, w których poprzednik v: $p[v] \in V'$



Złożoność: naiwny algorytm: $O(n^2)$

kopce Fibonacciego: $O(m + n \log n)$