Chapter 10

Graph connectivity

Discrete Structures for Computing

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le
Faculty of Computer Science and Engineering
University of Technology - VNUHCM
{htnguyen;trtanh}@hcmut.edu.vn

Acknowledgement

Some slides about Euler and Hamilton circuits are created by Chung Ki-hong and Hur Joon-seok from KAIST.

Course outcomes

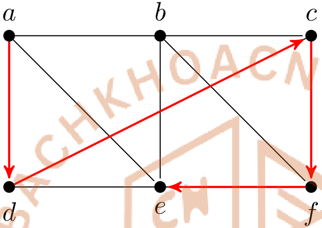
	Course learning outcomes
L.O.1	Understanding of logic and discrete structures
	L.O.1.1 – Describe definition of propositional and predicate logic L.O.1.2 – Define basic discrete structures: set, mapping, graphs
L.O.2	Represent and model practical problems with discrete structures
	L.O.2.1 – Logically describe some problems arising in Computing L.O.2.2 – Use proving methods: direct, contrapositive, induction L.O.2.3 – Explain problem modeling using discrete structures
L.O.3	Understanding of basic probability and random variables
	L.O.3.1 – Define basic probability theory L.O.3.2 – Explain discrete random variables
L.O.4	Compute quantities of discrete structures and probabilities
	L.O.4.1 – Operate (compute/ optimize) on discrete structures L.O.4.2 – Compute probabilities of various events, conditional ones, Bayes theorem

Contents

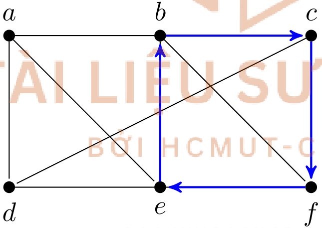
Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Contents
Connectivity
Paths and Circuits
Euler and Hamilton Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra's Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Ford's algorithm
Others
Graph Coloring

Paths and Circuits



Simple path of length 4

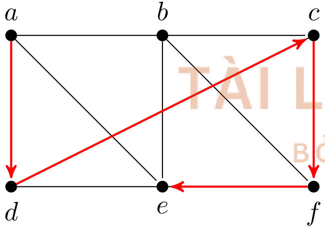


Simple circuit of length 4

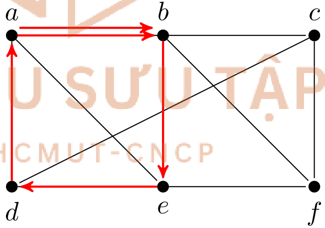
Path and Circuits

Definition (in undirected graph)

- Path (*đường đi*) of length n from u to v: a sequence of n edges $\{x_0, x_1\}, \{x_1, x_2\}, \dots, \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$.
- A path is a circuit (*chu trình*) if it begins and ends at the same vertex, $u = v$.
- A path or circuit is simple (*đơn*) if it does not contain the same edge more than once.



Simple path



Not simple path

Path and Circuits

Definition (in directed graphs)

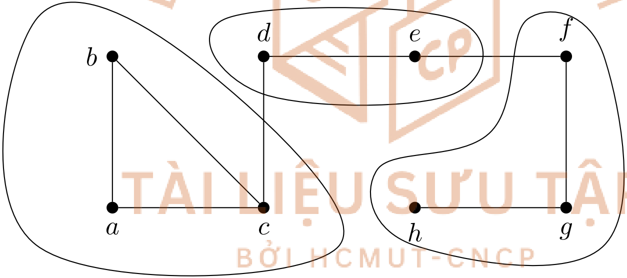
Path is a sequence of $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$, where $x_0 = u$ and $x_n = v$.

Connectedness in Undirected Graphs

Definition

- An undirected graph is called connected (*liên thông*) if there is a path between every pair of distinct vertices of the graph.
- There is a simple path between every pair of distinct vertices of a connected undirected graph.

planar : so mien = so canh - so dinh +2



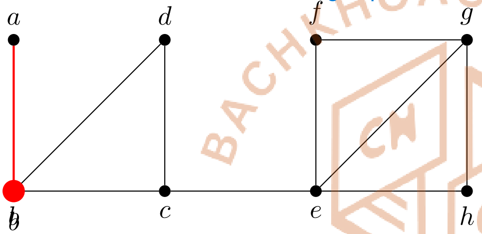
Connected graph
Disconnected graph
Connected components (*thành phần liên thông*)

How Connected is a Graph?

-A connected graph 'G' may have at most (n–2) cut vertices

-a cut edge e G if and only if the edge 'e' is not a part of any cycle in G

-the maximum number of cut edges possible is 'n-1' (n is num of vertices)



Definition

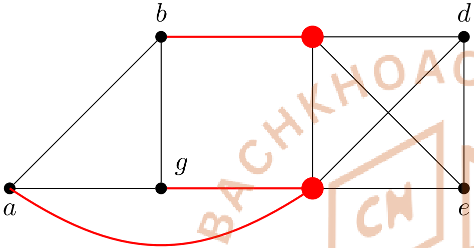
- b is a cut vertex (*đỉnh cắt*) or articulation point (*điểm khớp*). What else?
- $\{a, b\}$ is a cut edge (*cạnh cắt*) or bridge (*cầu*). What else?

moi quan he giua canh cat va dinh cat ?

- whenever cut edges exist, cut vertices also exist because at least one vertex of a cut edge is a cut vertex. cut edge => cut vertex
- cut vertex !=> cut edge

How Connected is a Graph?

+For any connected graph G
K(G) (G) (G)
Vertex connectivity (K(G)), edge connectivity ((G)), minimum number of degrees of G((G)).



Definition

- This graph doesn't have cut vertices: nonseparable graph (*đồ thị không thể phân tách*)
- The vertex cut is $\{c, f\}$, so the minimum number of vertices in a vertex cut, vertex connectivity (*liên thông đỉnh*) $\kappa(G) = 2$.
- The edge cut is $\{\{b, c\}, \{a, f\}, \{f, g\}\}$, the minimum number of edges in an edge cut, edge connectivity (*liên thông cạnh*) $\lambda(G) = 3$.

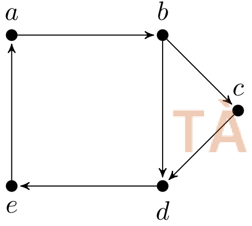
Applications of Vertex and Edge Connectivity

- Reliability of networks
  - Minimum number of routers that disconnect the network
  - Minimum number of fiber optic links that can be down to disconnect the network
- Highway network
  - Minimum number of intersections that can be closed
  - Minimum number of roads that can be closed

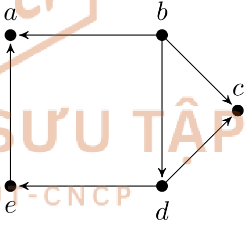
Connectedness in Directed Graphs

Definition

- An directed graph is strongly connected (*liên thông mạnh*) if there is a path between any two vertices in the graph (for both directions).
- An directed graph is weakly connected (*liên thông yếu*) if there is a path between any two vertices in the underlying undirected graph.



Strongly connected

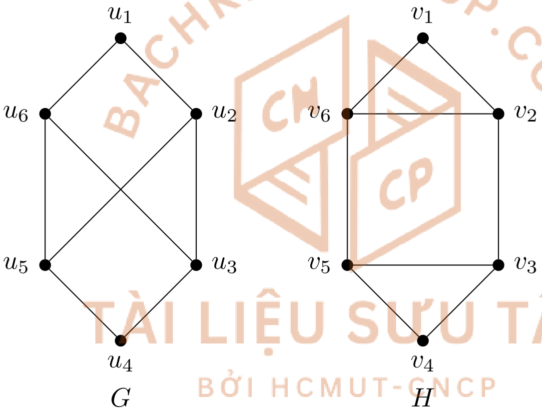


Weakly connected

Applications

Example

Determine whether the graphs below are isomorphic.



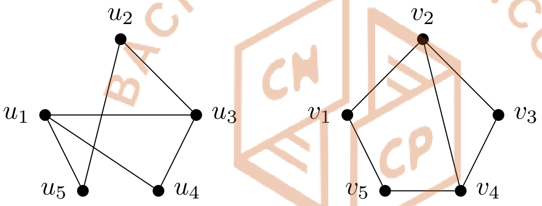
Solution

H has a simple circuit of length three, not G.

Applications

Example

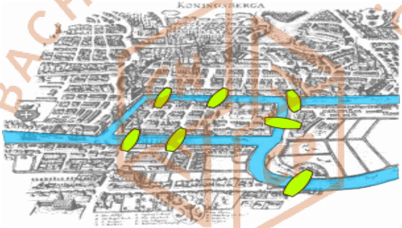
Determine whether the graphs below are isomorphic.



Solution

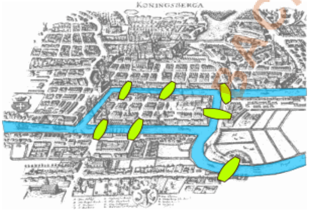
Both graphs have the same vertices, edges, degrees, circuits. They may be isomorphic.
To find a possible isomorphism, we can follow paths that go through all vertices so that the corresponding vertices in the two graphs have the same degrees.

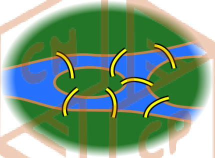
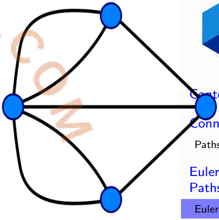
The Famous Problem of Seven Bridges of Königsberg



- Is there a route that a person crosses all the seven bridges once?

Euler Solution



- Euler gave the solution: It is not possible to cross all the bridges exactly once.

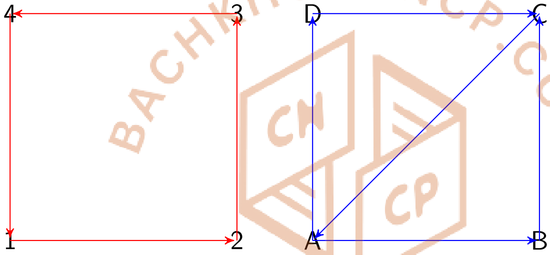
Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

What is Euler Path and Circuit?

- **Euler Path** (*đường đi Euler*) is a path in the graph that passes each edge only once.
  The problem of Seven Bridges of Königsberg can be also stated: Does Euler Path exist in the graph?
- **Euler Circuit** (*chu trình Euler*) is a path in the graph that passes each edge only once and return back to its original position.
  From Definition, Euler Circuit is a subset of Euler Path.

It must be a connected graph before being a Euler path/circuit

Examples of Euler Path and Circuit



Euler Circuit

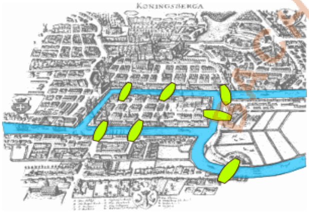
Euler Path

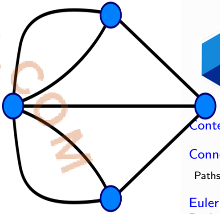
Conditions for Existence

In a connected multigraph,

- Euler Circuit existence: **no odd-degree nodes exist** in the graph.
- Euler Path existence: **2 or no odd-degree nodes exist** in the graph.

Back to the Seven Bridges Problem





- Four vertices of odd degree
- No Euler circuit $\rightarrow$ cannot cross each bridge exactly once, and return to starting point
- No Euler path, either

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Contents

Searching Euler Circuits and Paths – Fleury's Algorithm

- Choose a random vertex (if circuit) or an odd degree vertex (if path)
- Pick an edge joined to another vertex so that it is not a cut edge unless there is no alternative
- Remove the chosen edge. The above procedure is repeated until all edges are covered.

Searching Euler Circuits and Paths – Hierholzer's Algorithm

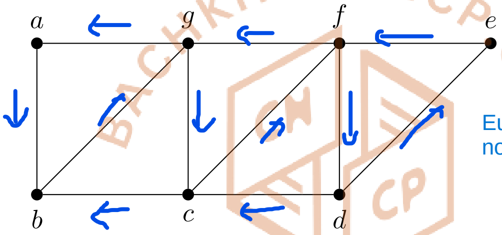
- Choose a starting vertex and find a circuit
- As long as there exists a vertex v that belongs to the current tour but that has adjacent edges not part of the tour, start another circuit from v

More efficient algorithm, $O(n)$.

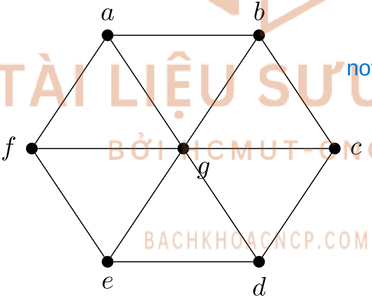
Exercise

Example

Are these following graph Euler path (circuit)? If yes, find one.

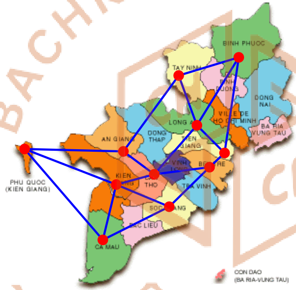


Euler path
not Euler circuit



not Euler path or circuit

Traveling Salesman Problem



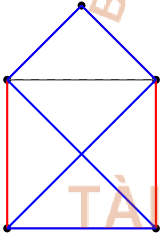
Is there the possible tour that visits each city exactly once?

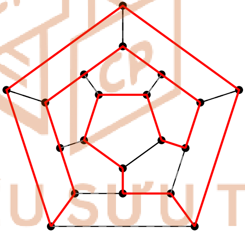
What Is A Hamilton Circuit?

K : n vertices , halminton <=> deg cua moi vertex > n/2

Definition

The circuit that visit each vertex in a graph once



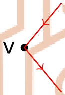


Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Rules of Hamilton Circuits

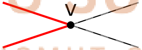
$\deg(v) = 2$ for $\forall v$ in Hamilton circuit!

Rule 1 if $\deg(v) = 2$, both edge must be used.



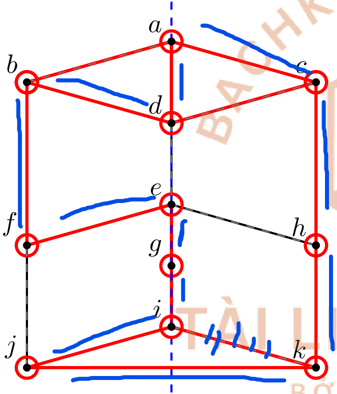
Rule 2 No subcircuit (*chu trình con*) can be formed.

Rule 3 Once two edges at a vertex v is determined, all other edges incident at v must be removed.



Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Finding Hamilton Circuits



We get Hamilton circuit!

Vertices : cities

Edges : possible routes

Rule 1

$\deg(v) = 2$

Rule 3

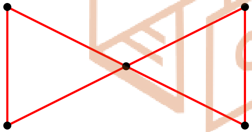
Once two edges are determined, other edges must be removed

Existence of Hamilton Circuit

condition : NO SUBcircuit

Hamilton circuit does not exist for all graph. But, there is no specific way to find whether Hamilton circuit exists or not.

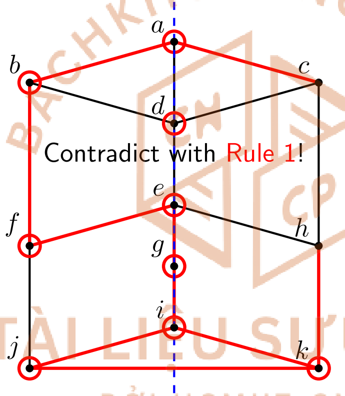
Simple check by rules of Hamilton circuit



Violates Rule 2! (No subcircuit)

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

We can verify nonexistence of the graph during find Hamilton circuit.



Hamilton circuit doesn't exist!

Application – Gray Code

Definition

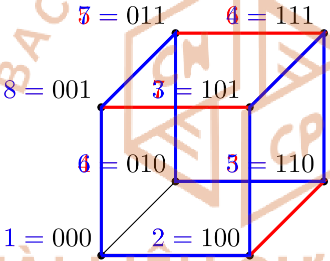
The binary sequence that express consecutive numbers by differing just one position of sequence.

Decimal number		Binary number	Gray code
1	=	001	000
2	=	010	100
3	=	011	110
4	=	100	010
5	=	101	011
$\vdots$		$\vdots$	$\vdots$

Used at digital communication for reduce the effect of noise; it prevents serious changes of information by noise.

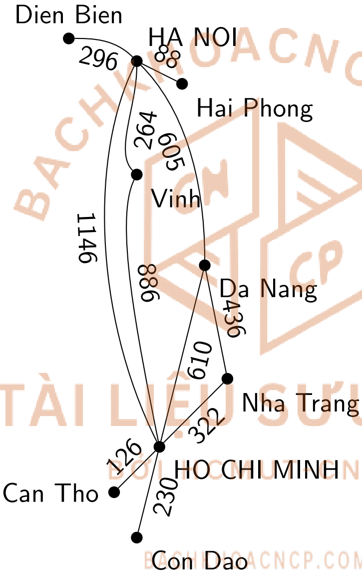
Gray Code

n-digit gray code can be generated by finding Hamilton circuits of n-dimensional hypercube! Consider the case $n = 3$.



Coordinate of each vertex is 3-digit binary sequences. Coordinates of adjacent vertices differ in just on place. Hamilton circuits of a cubic graph makes the order of binary sequences!

Weighted Graphs



Problem

The problem is also sometimes called the single-pair shortest path problem, to distinguish it from the following generalizations:

- The single-source shortest path problem, in which we have to find shortest paths from a source vertex v to all other vertices in the graph.
- The single-destination shortest path problem, in which we have to find shortest paths from all vertices in the graph to a single destination vertex v. This can be reduced to the single-source shortest path problem by reversing the edges in the graph.
- The all-pairs shortest path problem, in which we have to find shortest paths between every pair of vertices v, v' in the graph.

These generalizations have significantly more efficient algorithms than the simplistic approach of running a single-pair shortest path algorithm on all relevant pairs of vertices.

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

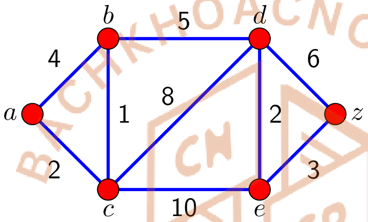
Dijkstra's Algorithm

```
procedure Dijkstra(G,a)
// Initialization Step
  forall vertices v
     Label[v] := ∞
     Prev[v] := -1
  endfor
  Label(a) := 0 // a is the source node
  S := ∅

// Iteration Step
  while z ∉ S
     u := a vertex not in S with minimal Label
     S := S ∪ {u}
     forall vertices v not in S
       if (Label[u] + Wt(u,v)) < Label(v)
         then begin
                Label[v] := Label[u] + Wt(u,v)
                Pred[v] := u
              end
  endwhile
```

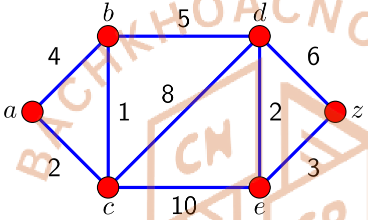

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Example



S	a	b	c	d	e	z
$\emptyset$	**0**	∞	∞	∞	∞	∞
a	0	4	**2**	∞	∞	∞
c	0	**3**	2	10	12	∞
b	0	3	2	**8**	12	∞
d	0	3	2	8	**10**	14
e	0	3	2	8	10	**13**

Example

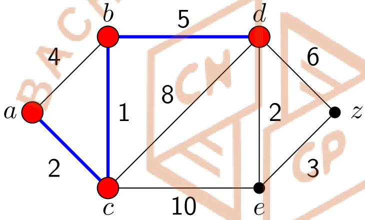


S	a	b	c	d	e	z
$\emptyset$	**0**	∞	∞	∞	∞	∞
a	0	4	2**2**	∞	∞	∞
c	0	3**3**	2	10	12	∞
b	0	3	2	8**8**	12	∞
d	0	3	2	8	10**10**	14
e	0	3	2	8	10	13**13**

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Back tracking procedure

How to determine shortest path from a to d according to Dijkstra's algorithm?



S	a	b	c	d	e	z
$\emptyset$	**0**	∞	∞	∞	∞	∞
a	**0**	4	**2**	∞	∞	∞
c	0	**3**	**2**	10	12	∞
b	0	**3**	2	**8**	12	∞
d	0	3	2	**8**	10	14
e	0	3	2	**8**	10	13

Dijkstra's Algorithm

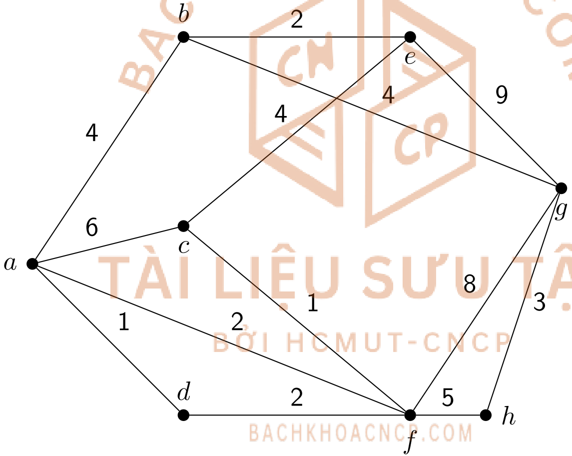
Property

Applicable for any G, any length $\ell(v_i) \geq 0$, $\forall i$; one-to-all; complexity $O(|V|^2)$.

Exercise

Example

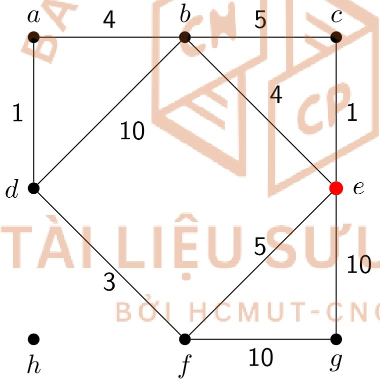
Find the shortest path from a to other vertices using Dijkstra's algorithm.



Exercise

Example

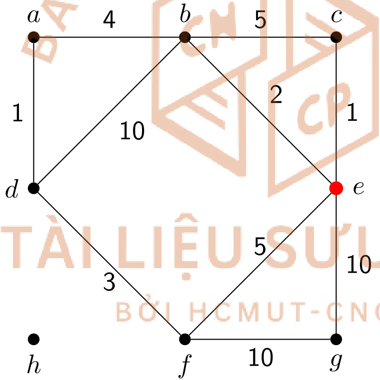
Find the shortest path from e to other vertices using Dijkstra's algorithm.



Exercise

Example

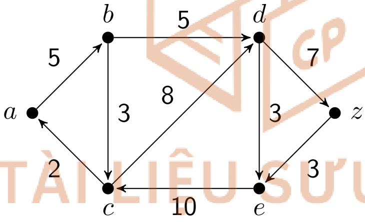
Find the shortest path from e to other vertices using Dijkstra's algorithm.



Exercise

Example

Find the shortest path from a to other vertices using Dijkstra's algorithm.

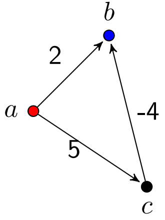


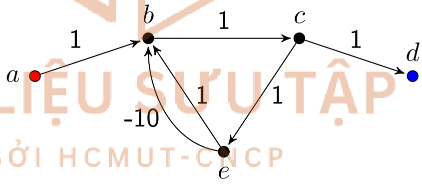
Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Dijkstra's Algorithm Flaw

Can Dijkstra's Algorithm be used on...

- ...digraph?
  - Yes!
- ...negative weighted graph?
  - No! Why?





Bellman-Ford Algorithm

```
procedure BellmanFord(G,a)
// Initialization Step
  forall vertices v
     Label[v] := ∞
     Prev[v] := -1
  Label(a) := 0 // a is the source node
// Iteration Step
  for i from 1 to size(vertices)-1
    forall vertices v
      if (Label[u] + Wt(u,v)) < Label[v]
        then
          Label[v] := Label[u] + Wt(u,v)
          Prev[v] := u
// Check circuit of negative weight
     forall vertices v
       if (Label[u] + Wt(u,v)) < Label(v)
         error "Contains circuit of negative weight"
```


Property

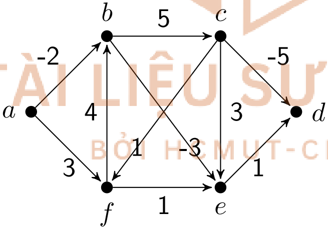
any G, any weighted; one-to-all; detect whether there exists a circle of negative weight; complexity $O(|V| \times |E|)$.

Example

Example

Step	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	-2a	∞	∞	∞	3a
2	0	-2	3b	∞	-5b	3
3	0	-2	3	-4e	-5	3
4	0	-2	3	-4	-5	3

Stop since Step 4 = Step 3.



Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

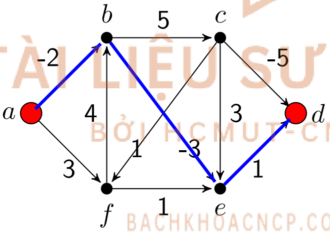
Backtracking procedure

Example

Step	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	**-2a**	∞	∞	∞	3a
2	0	**-2**	3b	∞	**-5b**	3
3	0	-2	3	**-4e**	**-5**	3
4	0	-2	3	**-4**	-5	3

Stop since Step 4 = Step 3.

How to find shortest path from a to d? $a \to b \to e \to d$

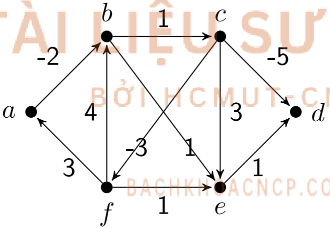


Example

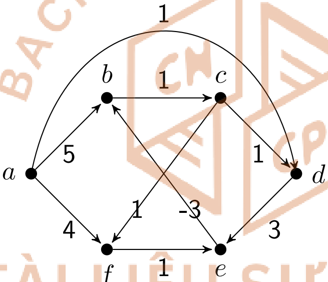
Example

Step	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	-2a	∞	∞	∞	∞
2	0	-2	-1b	∞	-1b	∞
3	0	-2	-1	-6c	-1	-4c
4	-1f	-2	-1	-6	-3f	-4
5	-1	-3a	-1	-6	-3	-4
6	-1	-3	-2b	-6	-3	-4
7	-1	-3	-2	-7c	-3	-4

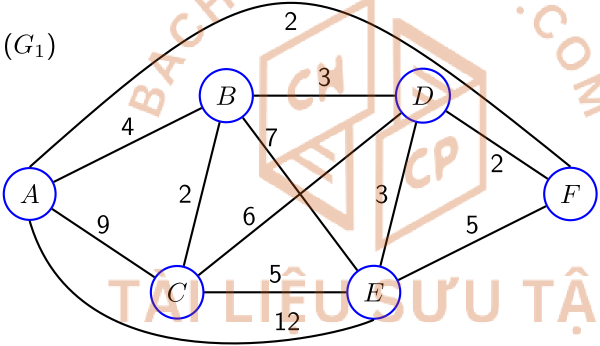
There exists a circle of negative weight since Step 6 $\neq$ Step 5.



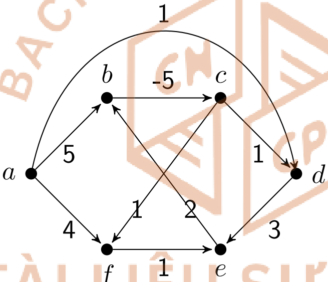
Exercise



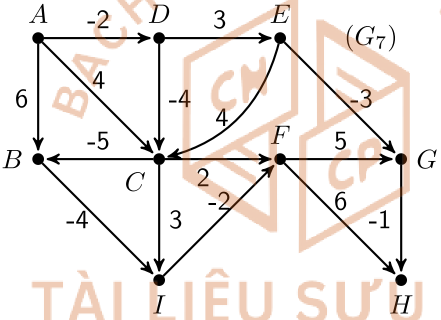
Exercise



Exercise



Exercise



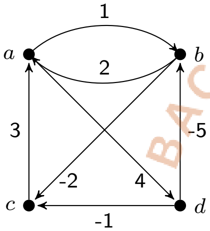
Floyd-Warshall Algorithm [1962]

```
procedure FloydWarshall ()
    for k := 1 to n
       for i := 1 to n
          for j := 1 to n
             path[i,j] = min (path[i,j],
                              path[i,k]+path[k,j]);
```

Property

any G, any weighted; all-to-all; this is an software algorithm; complexity $O(|V|^3)$.

Example



Shortest path from b to d
(5_3 from $L^{(4)}$):
$bd = bc + cd$
($5_3 = -2_0 + 7_1$ from $L^{(3)}$)
$cd = ca + ad$
($7_1 = 3_0 + 4_0$ from $L^{(1)}$)
$\Rightarrow bd = bc + ca + ad$

$$L^{(0)} = \begin{pmatrix} 0_0 & 1_0 & \infty_0 & 4_0 \\ 2_0 & 0_0 & -2_0 & \infty_0 \\ 3_0 & \infty_0 & 0_0 & \infty_0 \\ \infty_0 & -5_0 & -1_0 & 0_0 \end{pmatrix}$$

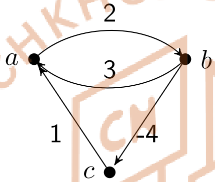
$$L^{(1)} = \begin{pmatrix} 0_0 & 1_0 & \infty_0 & 4_0 \\ 2_0 & 0_0 & -2_0 & 6_1 \\ 3_0 & 4_1 & 0_0 & 7_1 \\ \infty_0 & -5_0 & -1_0 & 0_0 \end{pmatrix}$$

$$L^{(2)} = \begin{pmatrix} 0_0 & 1_0 & -1_2 & 4_0 \\ 2_0 & 0_0 & -2_0 & 6_1 \\ 3_0 & 4_1 & 0_0 & 7_1 \\ -3_2 & -5_0 & -7_2 & 0_0 \end{pmatrix}$$

$$L^{(3)} = \begin{pmatrix} 0_0 & 1_0 & -1_2 & 4_0 \\ 1_3 & 0_0 & -2_0 & 5_3 \\ 3 & 4_1 & 0_0 & 7_1 \\ -4_3 & -5_0 & -7_2 & 0_0 \end{pmatrix}$$

$$L^{(4)} = \begin{pmatrix} 0_0 & -1_4 & -3_4 & 4_0 \\ 1_3 & 0_0 & -2_0 & 5_3 \\ 3_0 & 2_4 & 0_0 & 7_1 \\ -4_3 & -5_0 & -7_2 & 0_0 \end{pmatrix}$$

Example

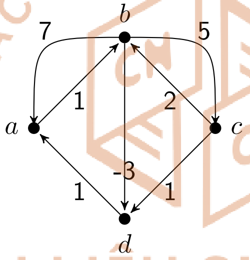


$$L^{(0)} = \begin{pmatrix} 0_0 & 2_0 & \infty_0 \\ 3_0 & 0_0 & -4_0 \\ 1_0 & \infty_0 & 0_0 \end{pmatrix} \quad L^{(1)} = \begin{pmatrix} 0_0 & 2_0 & \infty_0 \\ 3_0 & 0_0 & -4_0 \\ 1_0 & 3_1 & 0_0 \end{pmatrix}$$

$$L^{(2)} = \begin{pmatrix} 0_0 & 2_0 & -2_0 \\ 3_0 & 0_0 & -4_0 \\ 1_0 & 3_1 & {\color{red}-1_2} \end{pmatrix}$$

STOP, there exists a circuit of negative length.

Exercise



Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Ford's algorithm

$\pi(1) = 0$

For each $j \in V$ do

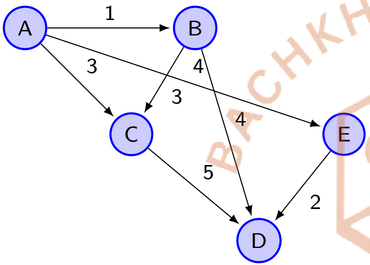
$\pi(j) = \min_{i \in \rho_j^{-1}} (\pi(i) + \ell[i,j])$

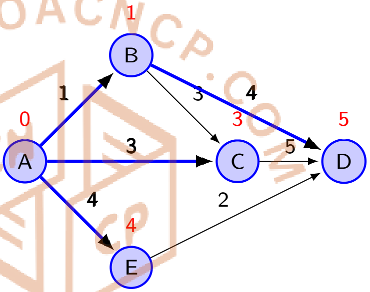
End

Property

G without circle, positive length; one-to-all; rank table definition; complexity $O(|V|)$.

Example

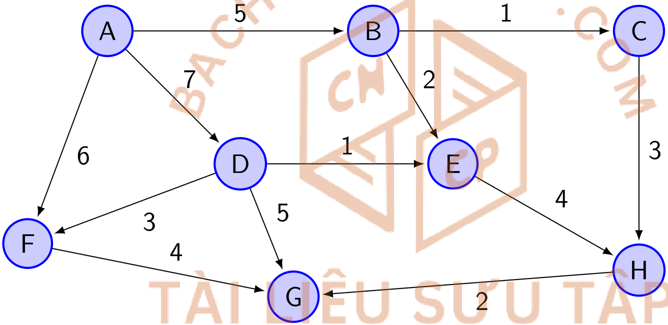




i	Γ_i^{-1}	rank(i)
A	-	0
B	A	1
C	A, B	2
D	B, C , E	3
E	A	1

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Exercise



Application

Problem

A young professor in Hue is invited to teach some years in Ho Chi Minh university of technology. He decides to represent the diverse operations of his transfer by a graph and, in this purpose, establishes the list of following operations:

- A: Find a house in Ho Chi Minh city.
- B: Choose a removal man and sign a contract of move
- C: Make pack his furniture by the removal man
- D: Make transport his furniture towards Ho Chi Minh city
- E: Find an accommodation to HCM (from Hue)
- F: Transport his family to HCM
- G: Move into his new accommodation
- H: Register the children to their new school
- I: Look for a temporary work for his wife
- J: Fit out the new accommodation and pay this arrangement with the first treatment of his wife
- K: Find a small bar to celebrate in family the success of the move and express the enjoyment to live in a good accommodation arrangement

Application

Considering constraint of posteriority following: $A < F$; $B < C$; $C < D \wedge F$; $D < G$; $E < F$; $F < G \wedge H \wedge I$; $G < K$; $H < K$; $I < J$; $J < K$.

Approximated task processing times :

A	B	C	D	E	F	G	H	I	J	K
10	2	3	4	7	3	5	1	3	8	2

Question

- Determine the minimal duration needed to completed all tasks.

Question

How to determine a shortest path from u to v in graph G which traverses at most $\leq$ a given constant number of intermediate vertices.

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Other shortest path problems

- multicriteria shortest path problem
  - linear combination
  - ε-constraint approach
  - lexico-graphical order
- k shortest paths problem
  - allowing loop
  - loopless
- multi-point shortest path
  - TSP, VRP

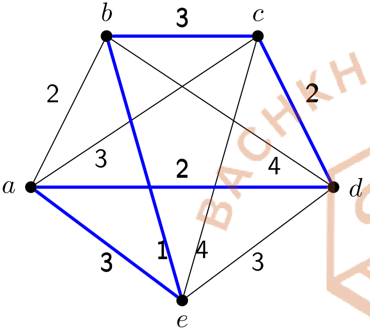
Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Traveling Salesman Problem (TSP)

Problem

- Given a set of n customers located in n cities and distances for each pair of cities, the problem involves finding a round-trip with the minimum traveling cost.
- The vehicle must visit each customer exactly once and return to its point of origin also called depot.
- The objective function is the total cost of the tour.
- $\mathcal{NP}$-complete: all known techniques for obtaining an exact solution require an exponentially increasing number of steps (computing resources) as the problems become larger.
- **TSP is one of the most intensely studied problems in computational mathematics, yet no effective solution method.**

Traveling Salesman Problem



- The total number of possible Hamilton circuit is $(n-1)!/2$.
- For example, if there are 25 customers to visit, the total number of solutions is $24!/2 = 3.1 \times 10^{23}$.
- If the depot is located at node 1, then the optimal tour is $1-5-2-3-4-1$ with total cost equal to 11.

Vehicle Routing Problem (VRP)

Problem

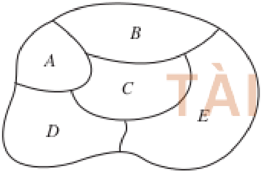
- The vehicle routing problem involves finding a set of trips, one for each vehicle, to deliver known quantities of goods to a set of customers.
- The objective is to minimize the travel costs of all trips combined.
- There may be upper bounds on the total load of each vehicle and the total duration of its trip.
- The most basic Vehicle Routing Problem (VRP) is the single-depot capacitate VRP.

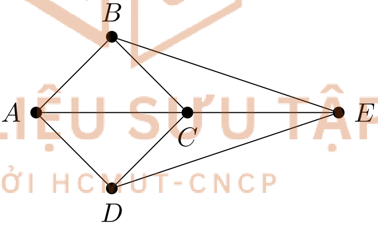
Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Maps and Graphs

Definition

- Every map can be represented by a graph. We call it dual graph.
- Problem of coloring the regions of a map $\rightarrow$ coloring the vertices of the dual graph so that no two adjacent vertices have the same color.

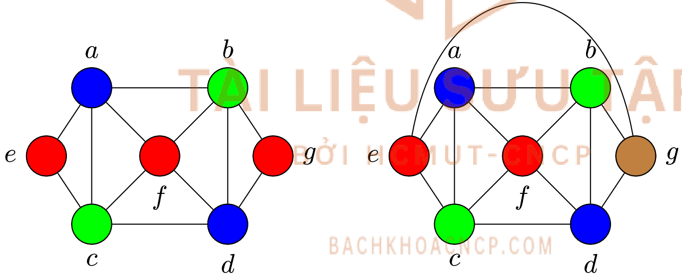




Graph coloring

Definition

- A coloring (*tô màu*) of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.
- The chromatic number (*số màu*) of a graph, denoted by $\chi(G)$, is the least number of colors needed for a coloring of this graph.



Four color theorem

Theorem (Four color theorem)

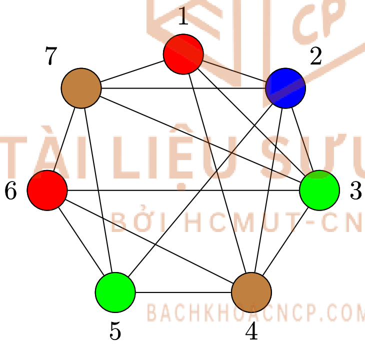
The chromatic number of a planar graph is no greater than four.

- Was a conjecture in the 1850s
- Was not proved completely until 1976 by Kenneth Appel and Wolfgang Haken, using computer
- No proof not relying on a computer has yet been found

Applications of Graph coloring

Scheduling Final Exam

- How can the final exams at a university be scheduled so that no student has two exams at the same time?
- Suppose we have 7 finals, numbered 1 through 7.
- The pairs of courses have common students are depicted in the following graph



Applications of Graph Coloring

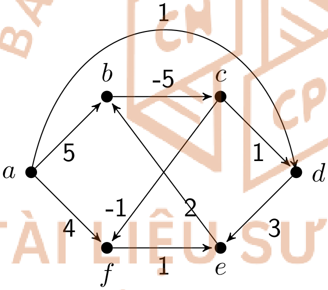
Other Applications

- Frequency Assignments: Television channels 2 through 12 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel. How can the assignment of channels be modeled by graph coloring?
- Index Registers: In an execution of loop, the frequently used variables should be stored in index registers to speed up. How many index registers are needed?

Huynh Tuong Nguyen, Tran Tuan Anh, Nguyen Ngoc Le

Exercise

Determine a shortest path from a to other vertices in the following graph.



Contents