Application Problems on Graphs and Trees

Chapter 12, 13

부산대학교
PUSAN NATIONAL UNIVERSITY

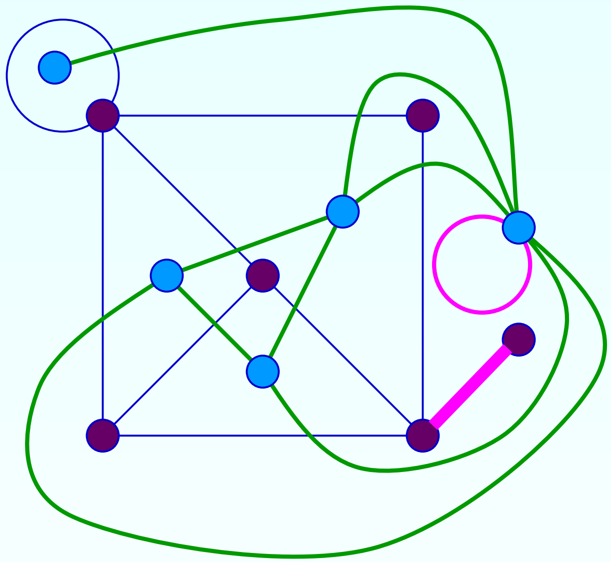
Dual Graphs

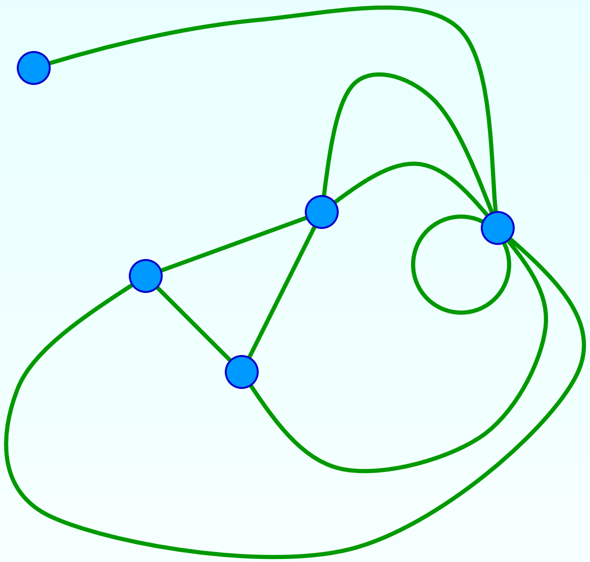
- Valid for planar graphs or multigraphs

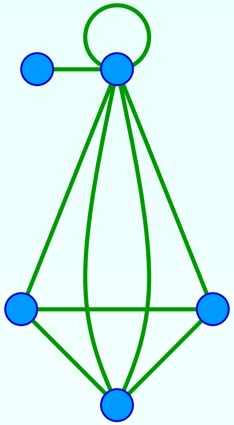
Procedure:

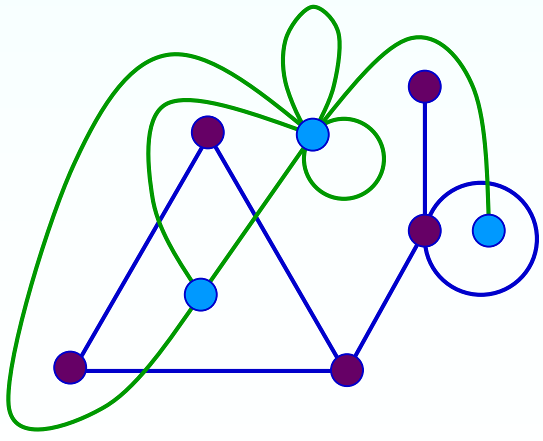
1. Place a vertex inside each region
2. For each edge shared by two regions, draw an edge connecting the vertices inside these regions
3. For an edge that is counted twice in computing the degree of a region, draw a loop at the vertex for the region

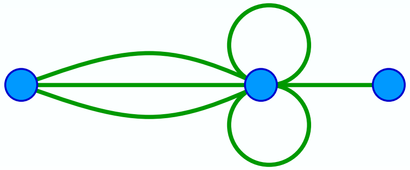
Examples of Dual graphs



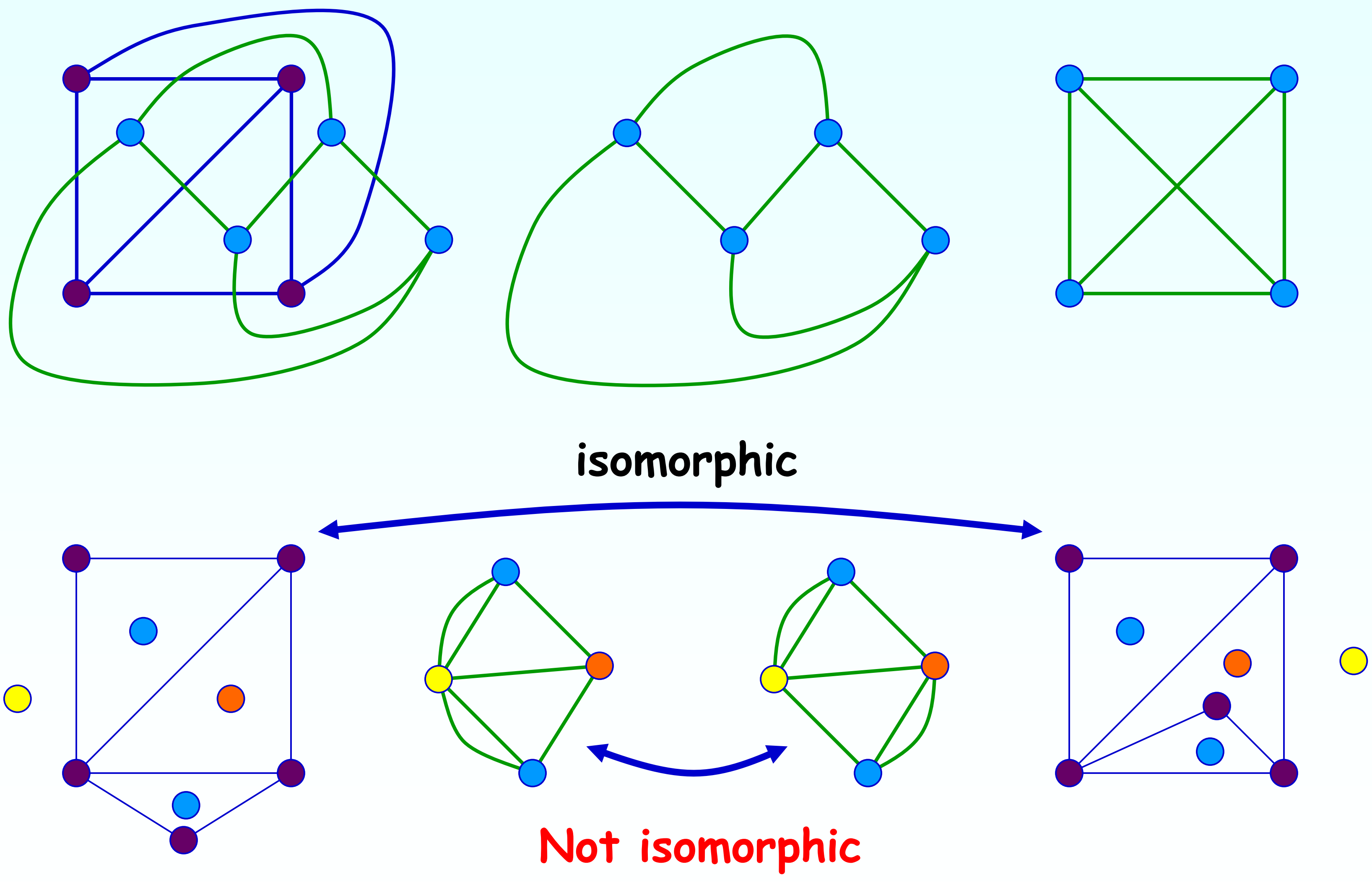








Examples of Dual graphs

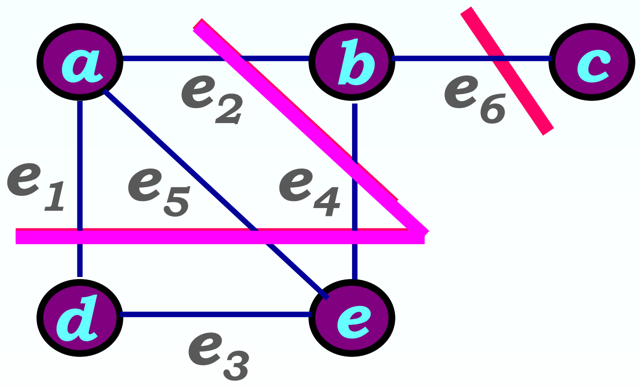


Cut-set

Let $G = (V,E)$ be an undirected graph.

A subset E' of E is called a cut-set of G if by removing the edges (but not vertices) in E' from G, we have $\kappa(G) < \kappa(G')$, where $G' = (V, E\text{-}E')$;

However, when we remove any *proper subset* E'' of E', we should have $\kappa(G) = \kappa(G'')$ for $G'' = (V, E - E'')$



Cut-set

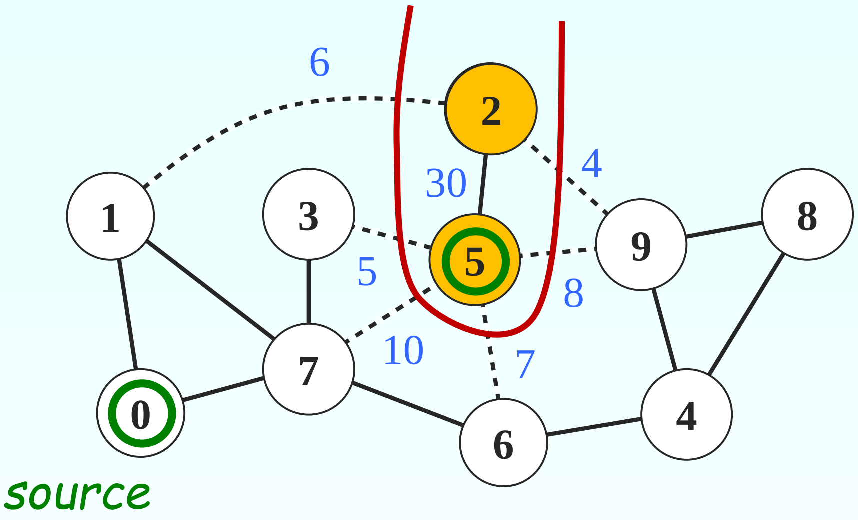
; {e_2, e_4}
; {e_1, e_4, e_5}
; {e_6} **bridge**

Not cut-set

; {e_2}
; {e_1, e_2, e_4, e_5}

$\kappa(G)$: # of connected components in G

Application



A cut = (S,T)
S = {2,5}
T = {0,1,3,4,6,7,8,9}

The cut-set of (S,T) =
{{2,1},{5,3},{5,7},{5,6},{5,9},{2,9}}

Max flow : 40 (= 6+5+10+7+8+4)

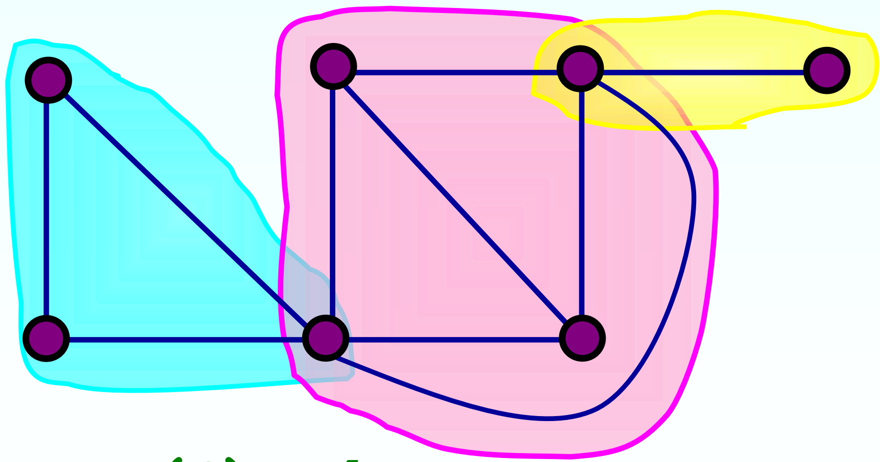
Max-flow min-cut theorem (in section 13.3)

The maximum network flow and the sum of the cut-edge weights of any minimum cut that separates the *source* and the *sink* are equal

Clique

If $G = (V,E)$ is an undirected graph, any subgraph of G that is a complete graph is called a clique in G.

The number of vertices in a largest clique in G is called the clique number for G and is denoted by $\omega(G)$.



$\omega(G) = 4$

Chromatic Number $\geq$ Clique Number

$\chi(G) \geq \omega(G)$

An NP-complete problem

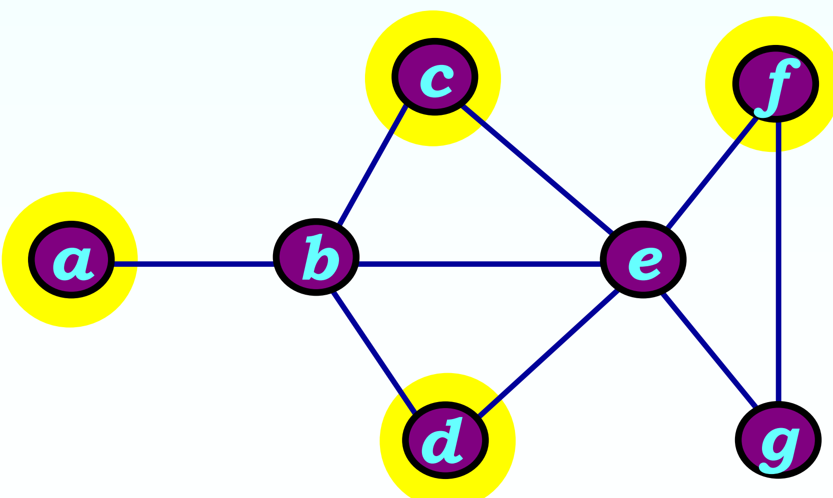
Clique(도당, 파벌): people(vertex), edge(relation “know”)

Independent Set

If $G = (V,E)$ is an undirected graph, a subset I of V is called **independent** if no two vertices in I are adjacent

An independent set I is called **maximal** if no vertex v can be added to I with $I \cup \{ v \}$ independent

The size of a largest independent set is called the **independence number** of G, denoted $\beta(G)$



Independent sets

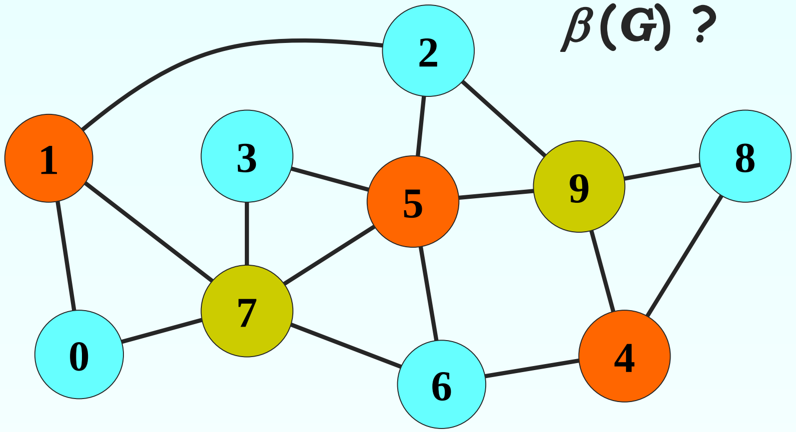
; {a, f}

; {a, c, g}

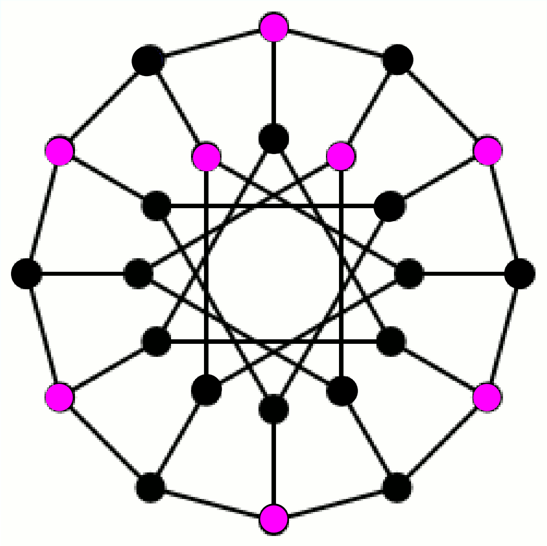
A maximal independent set

; {a, c, d, f}

Application

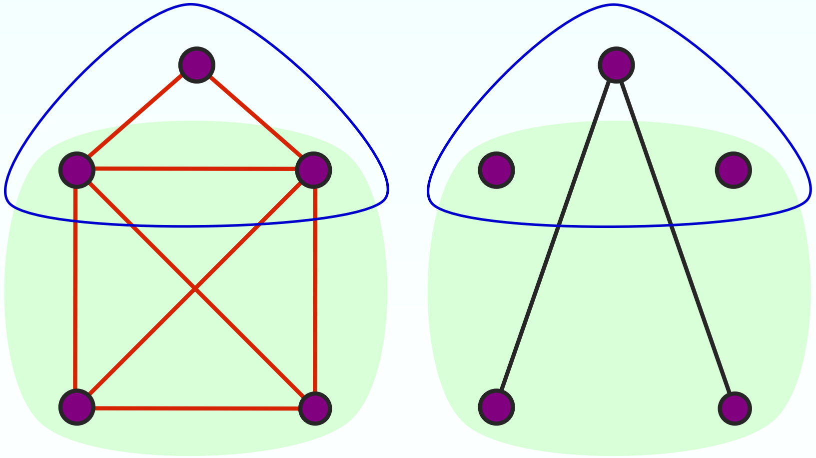


$\beta(G)$?

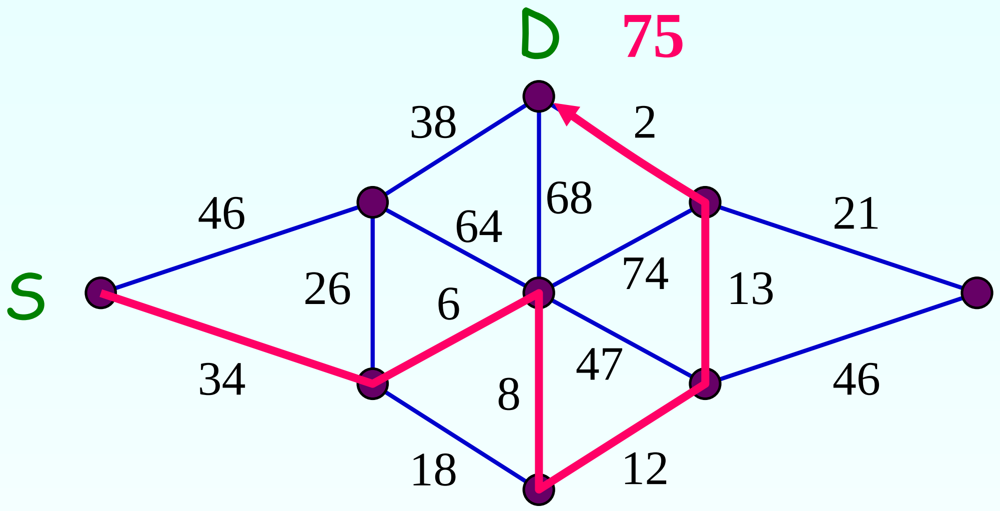


An NP-complete problem

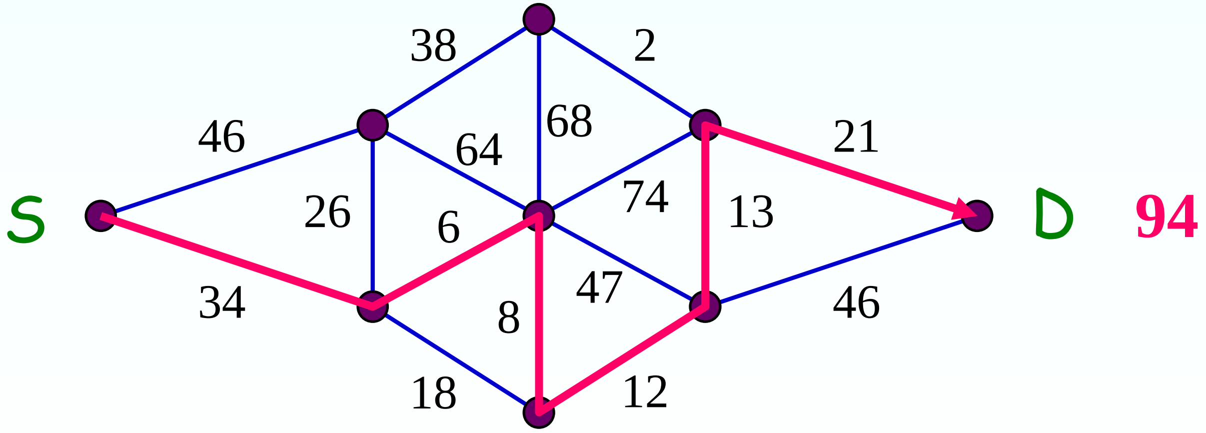
If a graph has an independent set of size k, then its complement has a clique of size k



Shortest Path

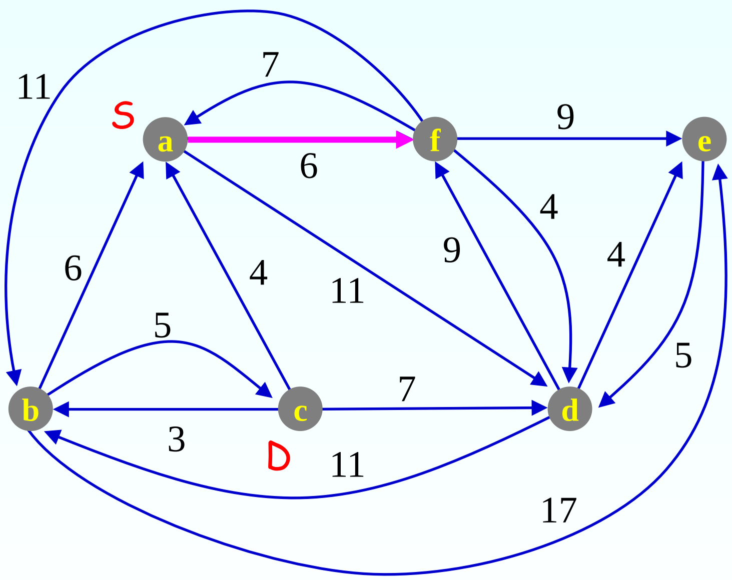


in Navigation Applications



Shortest Path

Dijkstra's Algorithm
(in section 13.1)



Initialization $S_0 = \{a\}$

a	f	e
(0,-)	(∞,-)	(∞,-)

b	c	d
(∞,-)	(∞,-)	(∞,-)

Cost from S

Previous node in S_0

1st Iteration $S_1 = \{a,f\}$

a	f	e
(0,-)	(6,a)	(∞,-)

b	c	d
(∞,-)	(∞,-)	(11, a)

$\$(d) = \min \{ \infty, \$(a) + w(a,d) \}$

$= \min \{ \infty, 0+11 \} = 11$

for all elements in S_0

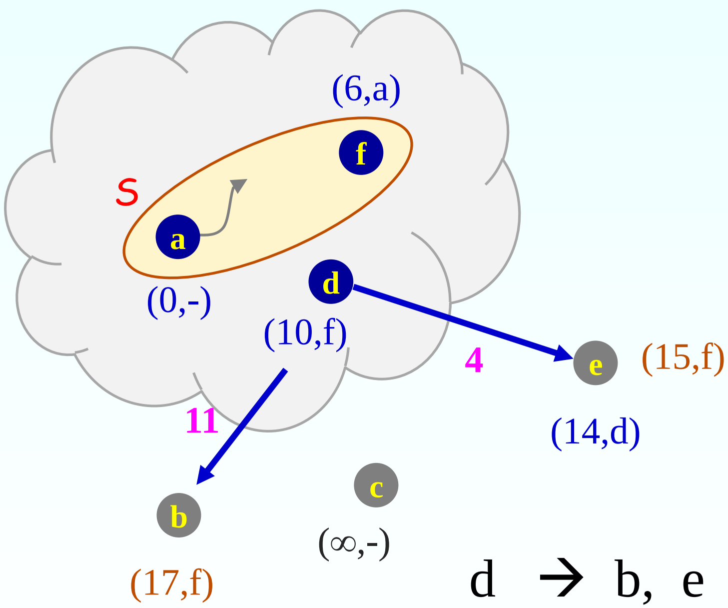
2nd Iteration ?

f → b, d, e

17, 10, 15

Shortest Path

Dijkstra's Algorithm



d → b, e

17, 14

2nd Iteration S_2 = {a,f,d}

a (0,-) f (6,a) e (15,f)

b (17,f) c (∞,-) d (10,f)

3rd Iteration S_3 = {a,f,d,e}

a (0,-) f (6,a) e (14,d)

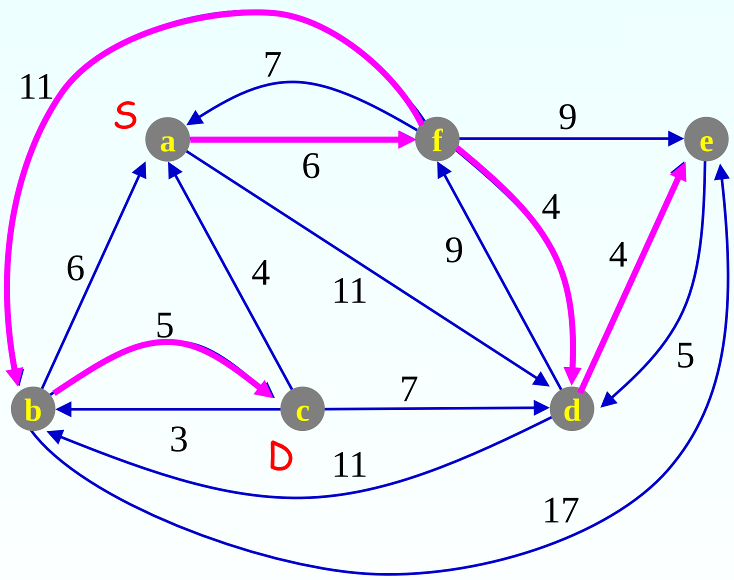
b (17,f) c (∞,-) d (10,f)

$(e) = min { 15, $(d) + w(d,e) }

= min { 15, 10+4 } = 14

Shortest Path

Dijkstra’s Algorithm



$\therefore$ a → f → b → c

4^{th} Iteration

S_4 = {a,f,d,e,b}

a ● (0,-)   f ● (6,a)   e ● (14,d)

b ● (17,f)   c ● (∞,-)   d ● (10,f)

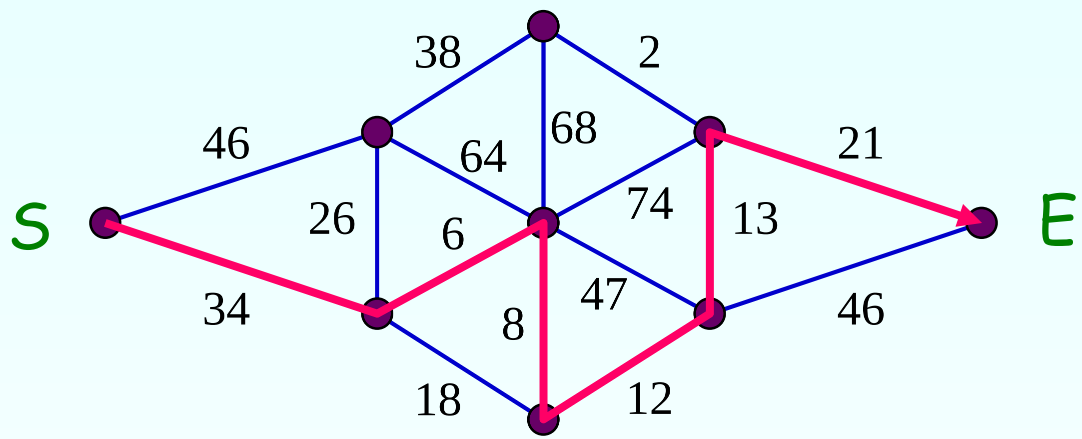
5^{th} Iteration

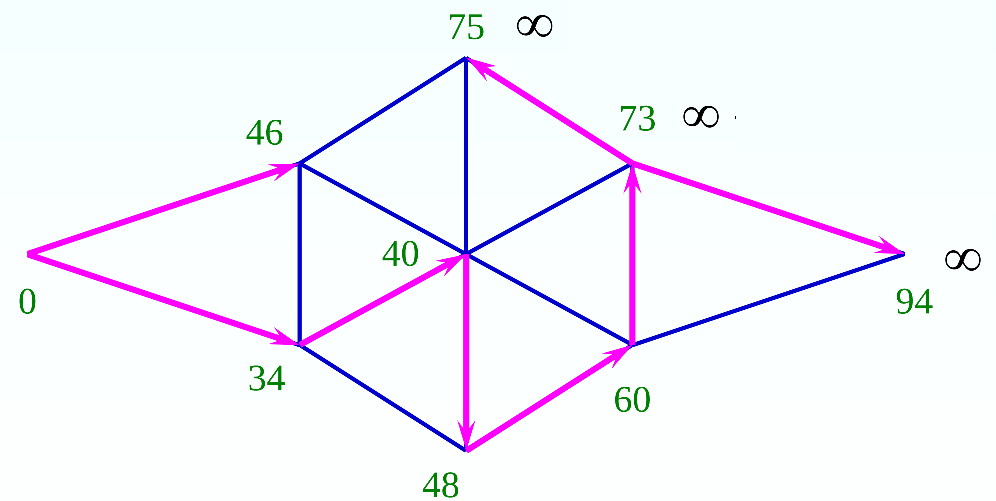
S_5 = {a,f,d,e,b,c}

a ● (0,-)   f ● (6,a)   e ● (14,d)

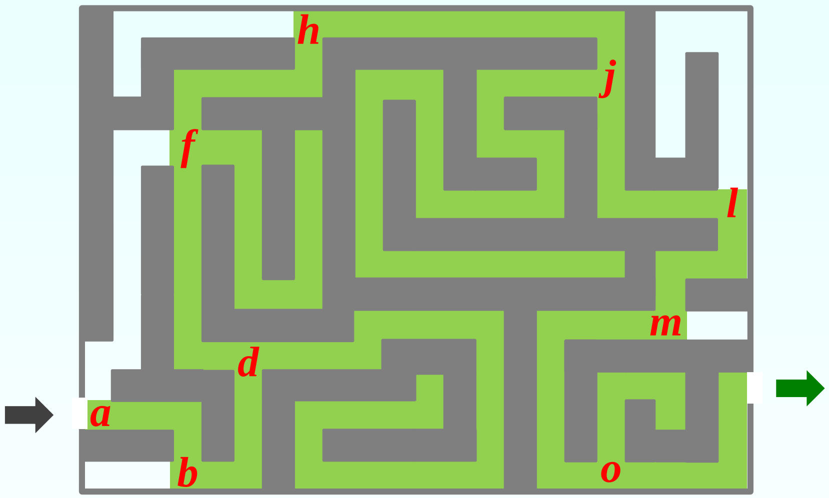
b ● (17,f)   c ● (22,b)   d ● (10,f)

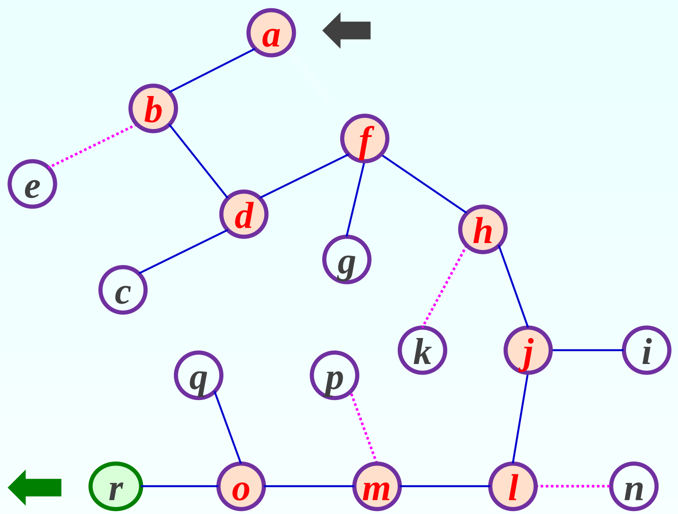
Shortest Path





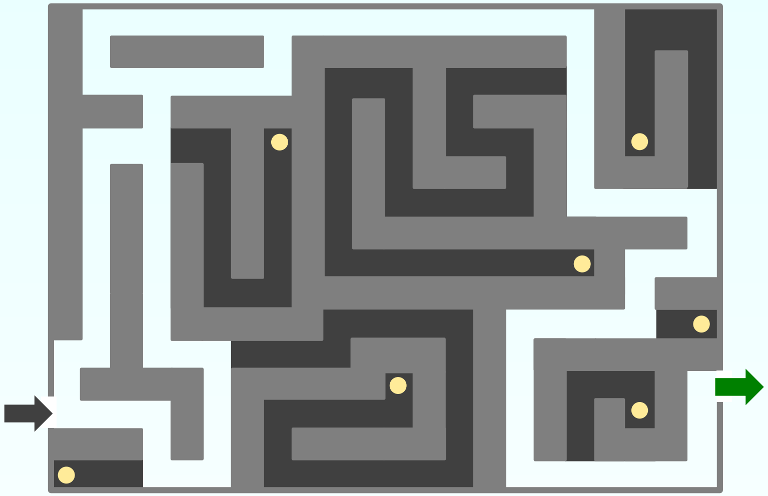
Graph Search



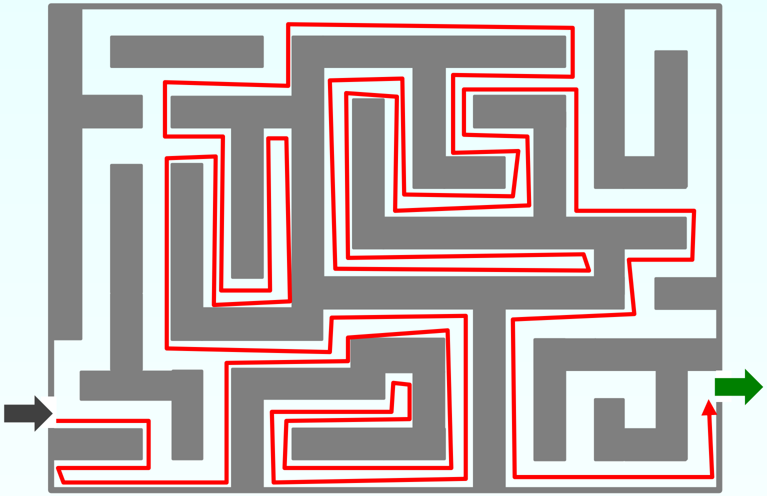


Applications: maze solving, web-crawling, spanning tree finding, connected component finding, topological sorting, etc.

Map (○)



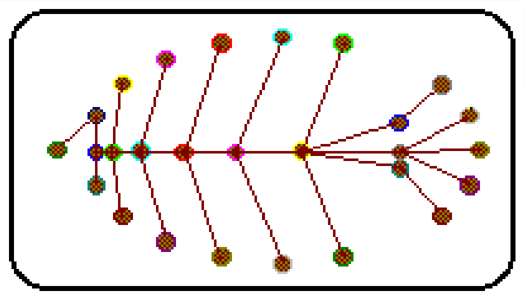
Map (×)

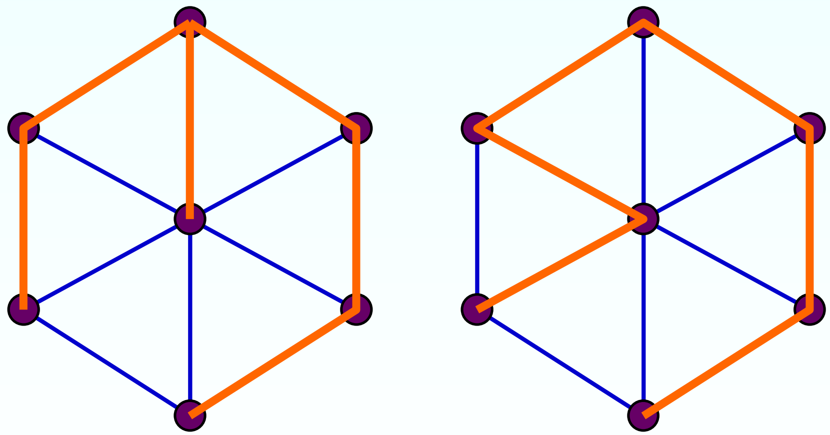


Graph Search

Input: A loop-free undirected graph $G = (V,E)$ where $|V| = n$ and the vertices are ordered as $v_1, v_2, \ldots, v_n$

Output: A spanning tree T for the specified order





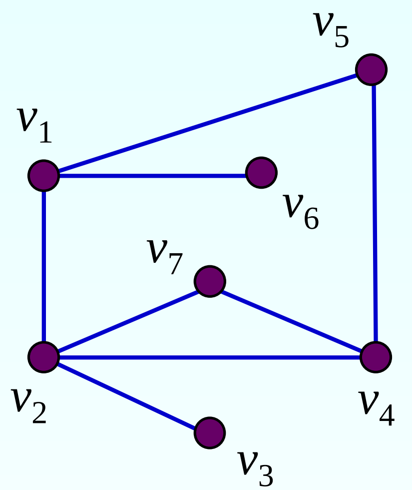
Wheel graph

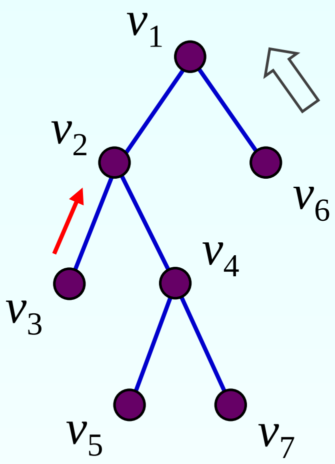
A spanning tree for a connected graph G is a *connected subgraph* of G that is a *tree* and contains *all the vertices* of G.

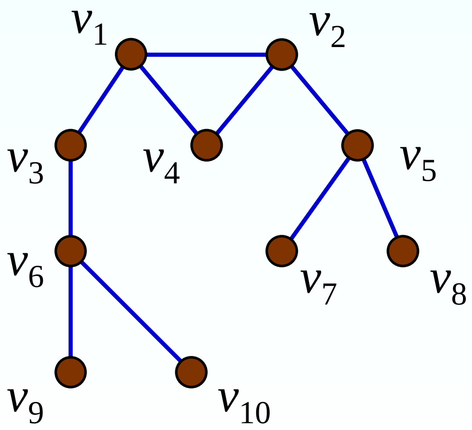
Graph Search

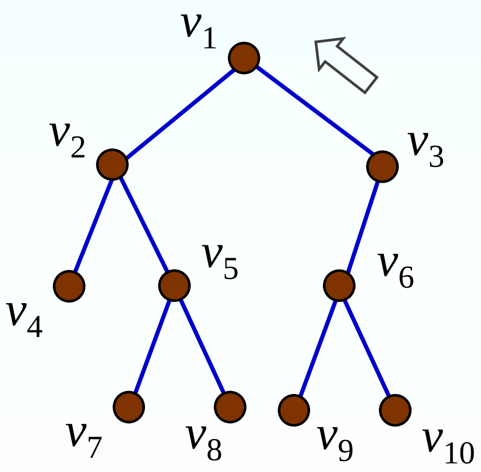
- **Depth-First Search Algorithm**
  - Assign v_1 to the variable v
  - Select the smallest subscript i, for $2 \le i \le n$, such that $\{v, v_i\} \in E$ and v_i has not already been visited
    - Attach $\{v, v_i\}$ to T and visit v_i. Assign v_i to v.
    - Repeat this step
  - If no such subscript is found, backtrack from v to its parent u in T. Assign u to v and return to the above step
  - If backtracking is not allowed any more, T is the spanning tree for the specified order

Graph Search

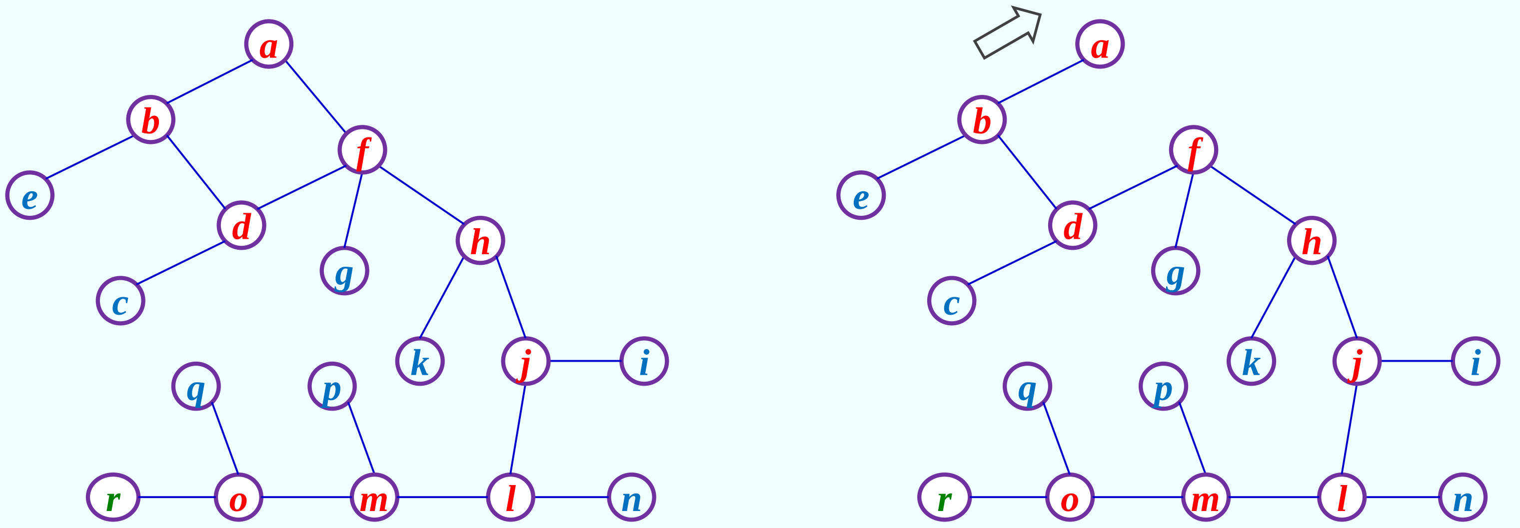




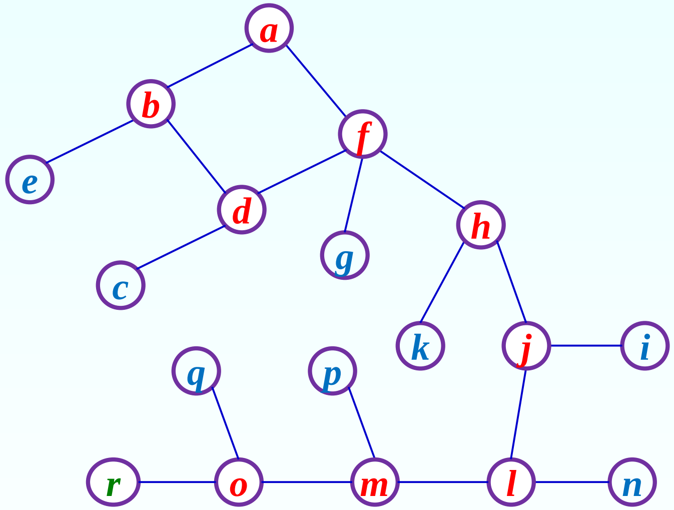


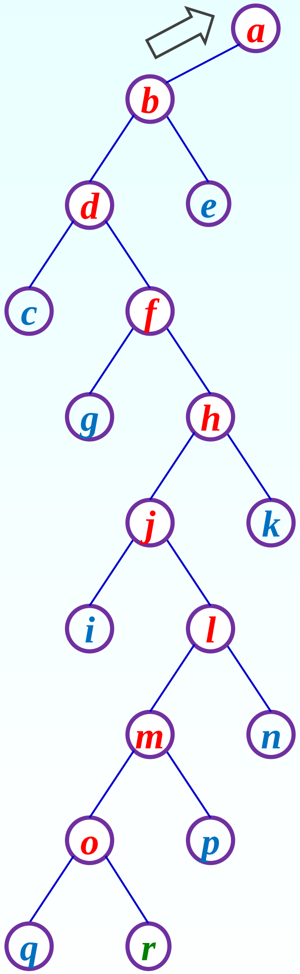


Graph Search



Graph Search

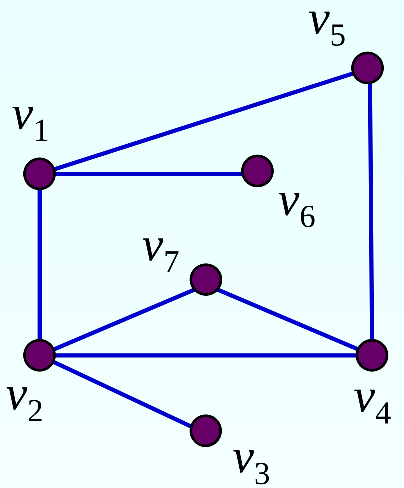


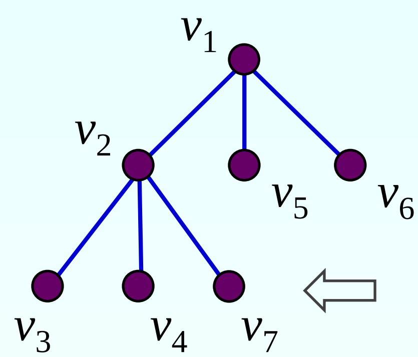


Graph Search

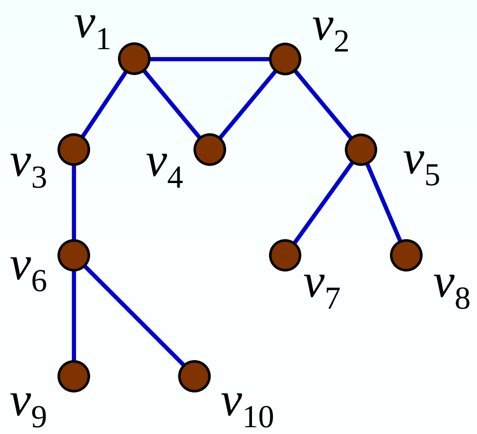
- **Breadth-First Search Algorithm**
  - Insert v_1 to an empty queue Q and initialize T as the tree made up of this vertex v_1. Visit v_1
  - While Q is not empty, delete a vertex v from Q. Now examine the vertices v_i (for $2 \leq i \leq n$) that are adjacent to v – in the specified order. If v_i has not been visited, perform the following: (1) Insert v_i to Q; (2) Attach the edge $\{v, v_i\}$ to T; (3) Treat v_i as a visited vertex
  - If we examine all of the vertices in Q and obtain no new edges, then T is the spanning tree for the order specified

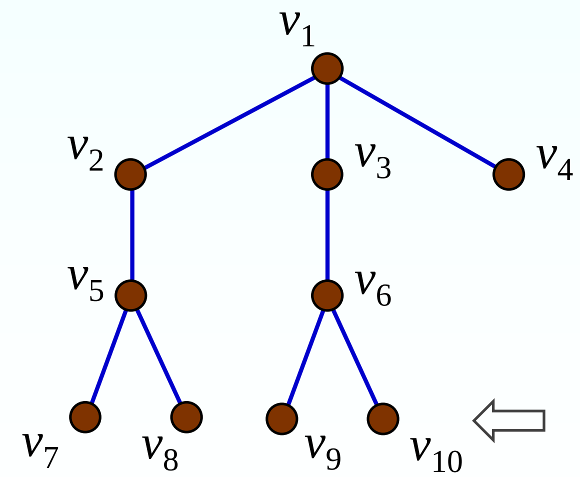
Graph Search



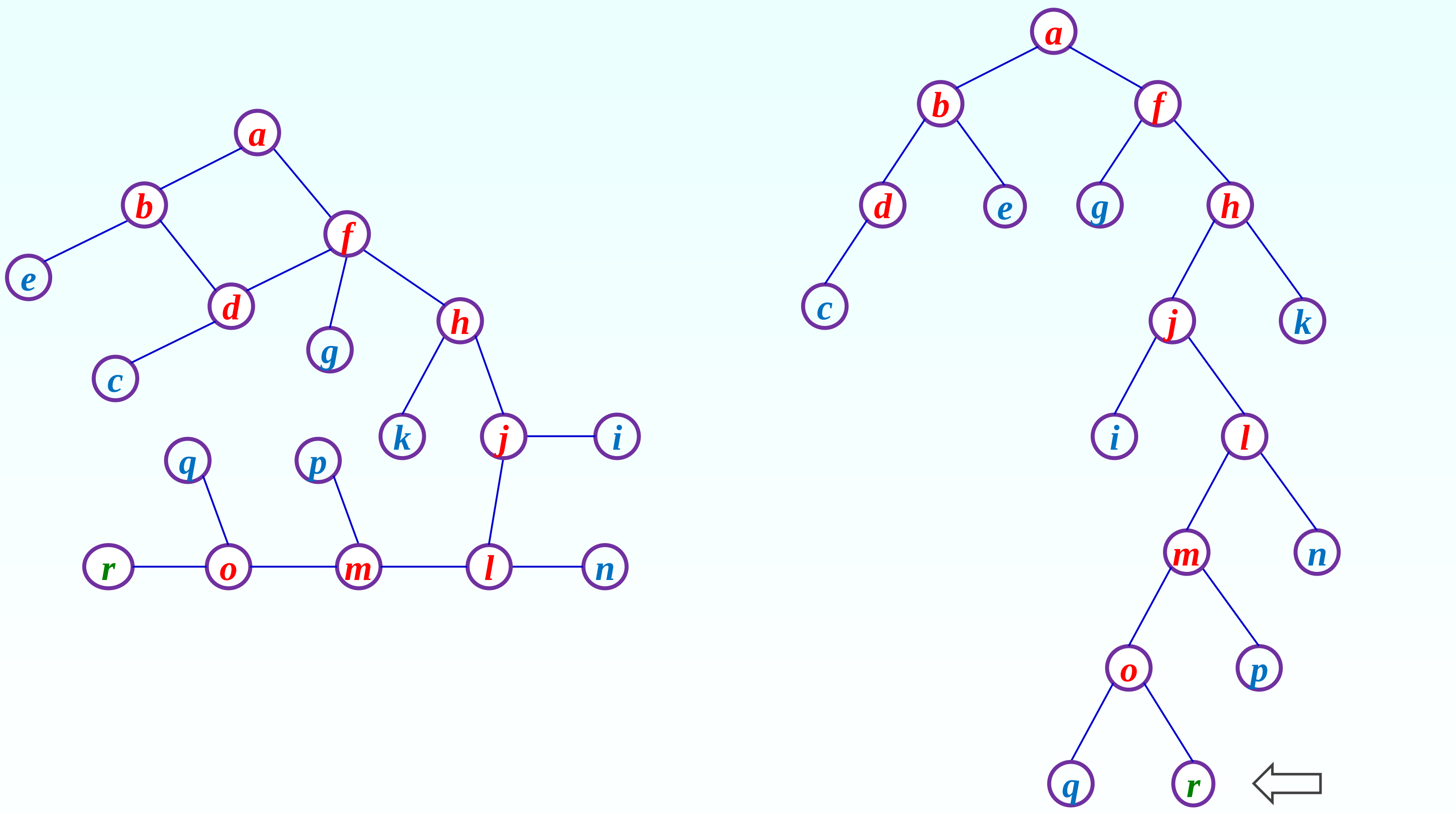


Queue



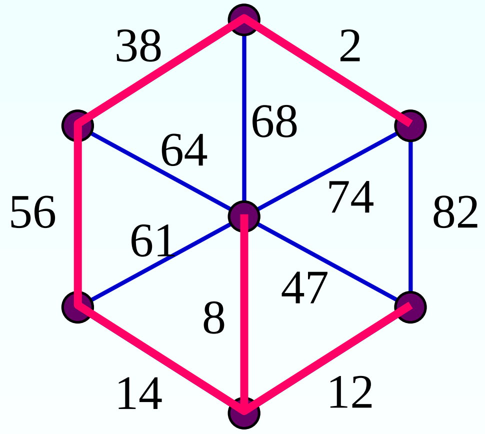


Graph Search



Minimum Spanning Tree

A spanning tree for a weighted graph is called **a minimum spanning tree**, if it has the minimal sum of weights.



A weighted graph

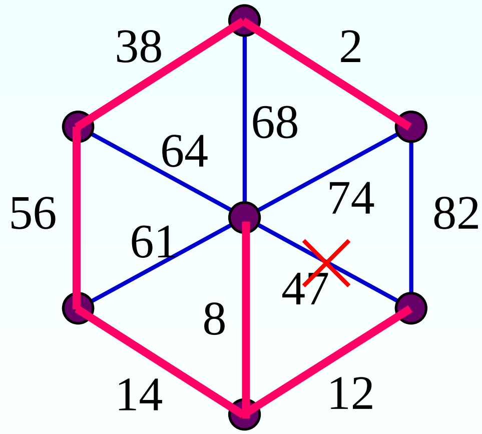
Sum of weights
130 (= 2+38+56+14+8+12)

Is it minimum ?

Minimum Spanning Tree

Kruskal's Algorithm
(in section 13.2)

Select an edge with the weight as small as possible unless it forms a cycle



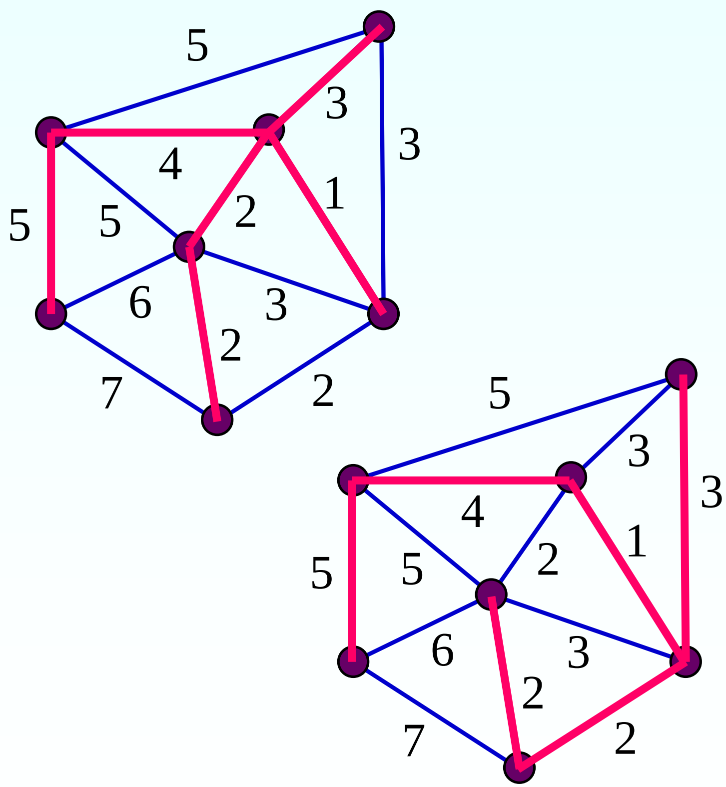
Weighted graph

Prim's Algorithm

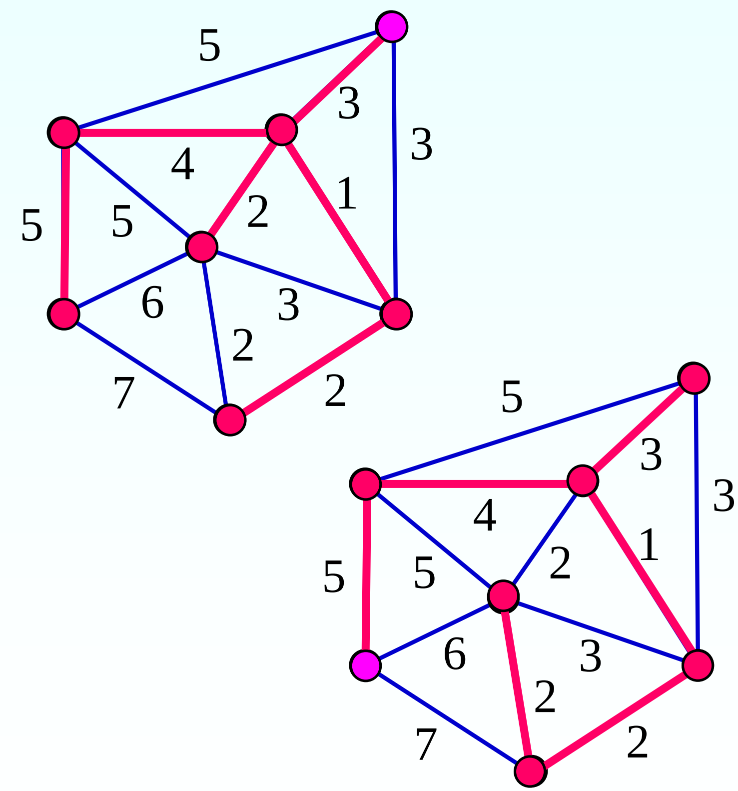
Add to T a shortest edge that connects one of visited vertices with an unvisited vertex u. Treat u as a visited vertex

Minimum Spanning Tree

Kruskal's Algorithm

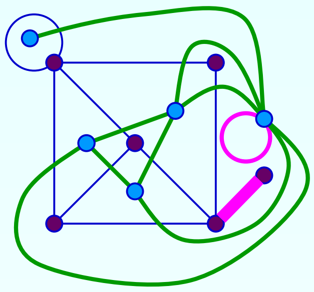


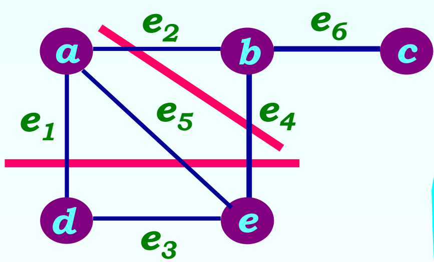
Prim's Algorithm

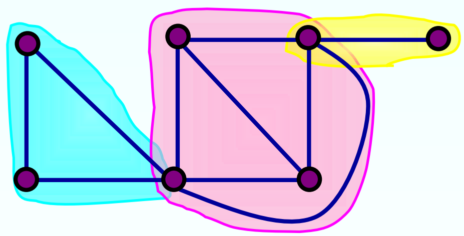


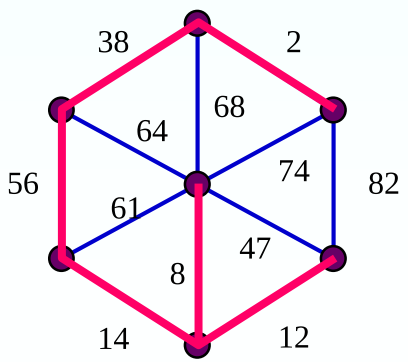
Summary of Miscellanies

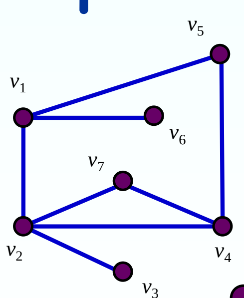
- Dual Graph for Planar Graphs
- Cut-Set
- Clique, Clique Number $\omega(G) \le \chi(G)$
- Independent Set
- Shortest Path
- Graph Search
- Minimum Spanning Tree

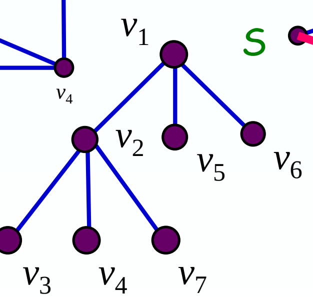


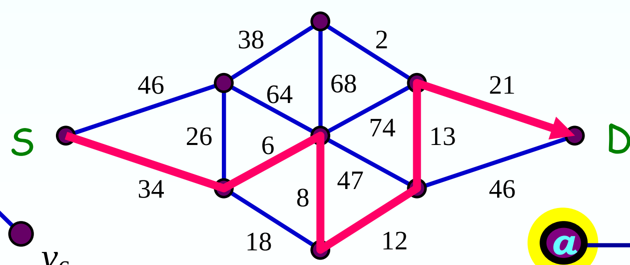


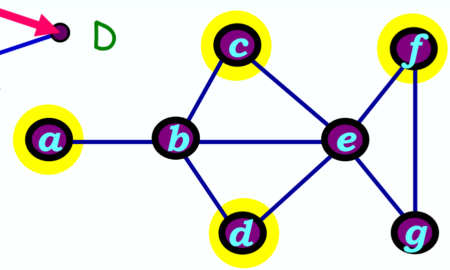












HW #1

- Exercises in Chapter 11
  - 11.1 : 8, 12, 14
  - 11.2 : 6, 8, 12
  - 11.3 : 4, 8, 18, 32 up to isomorphism ?
  - 11.4 : 14, ~~22~~
  - 11.5 : 12, 24
  - 11.6 : 7