VE203
Discrete Math
RC6

Yucheng Huang

University of Michigan
Shanghai Jiao Tong University
Joint Institute

April 15, 2022

Definition

Graph

A graph G consists of a set of vertices, denoted by $V(G)$, a set of edges, denoted by $E(G)$, and a relation called incidence so that each edge is incident with either one or two vertices, called ends (or endpoints). For convenience, we sometimes write $G = (V, E)$ to indicate that G is a graph with vertex set V and edge set E.

Adjacent

Two distinct vertices u, v in a graph G are adjacent if there is an edge with ends u, v. We also call u, v neighbors in G.

Loops, Parallel Edges, and Simple Graphs

An edge with just one end is called a loop. Two distinct edges with the same ends are are parallel (called "parallel edges" or "multiple edges"). A graph without loops or parallel edges is called simple.

Graph Definition

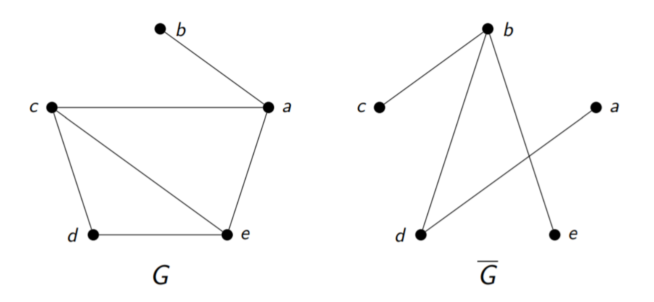
Isomorphism

An isomorphism from a simple graph G to a simple graph H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$. We say "G is isomorphic to H'', donoted $G \cong H$, if there is an isomorphism from G to H.

Graph Definition

Complement

The complement $\bar{G}$ of a simple graph G is the simple graph with vertex set $V(G)$ defined by $uv \in E(\bar{G})$ iff $uv \notin E(G)$. Note that given graph $G = (V, E)$, we have $\bar{G} = \left(V, \binom{V}{2} - E\right)$.



Graph Definition

The null graph

The null graph is the graph whose vertex set and edge set are empty.

Clique

A clique in a graph is a set of pairwise adjacent vertices.

SubGraph

If G, H have $V(H) \subset V(G)$, and $E(H) \subset E(G)$ with incidence in H the same as G, then H is a subgraph of G, denoted by $H \subset G$.
Obviously, given $H_1, H_2 \subset G$, then $H_1 \cap H_2 \subset G$, with

$$V(H_1 \cap H_2) = V(H_1) \cap V(H_2)$$
$$E(H_1 \cap H_2) = E(H_1) \cap E(H_2)$$

$H_1 \cup H_2 \subset G$, with

$$V(H_1 \cup H_2) = V(H_1) \cup V(H_2)$$
$$E(H_1 \cup H_2) = E(H_1) \cup E(H_2)$$

Degree

Definition

The degree of a vertex v in a graph G, denoted $\deg(v)$ is the number of incident edges (loops counted twice). We write $\deg_G(v)$ if G is not clear (i.e., we are not sure if $v \in V(G)$).

Theorem

For all $G = (V, E)$,

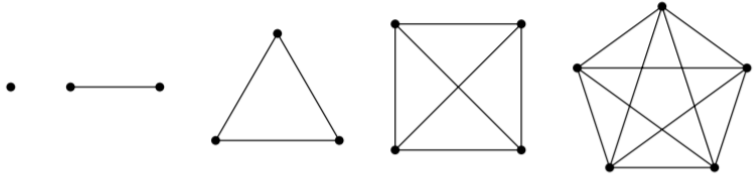
$$\sum_{v \in V} \deg(v) = 2|E|$$

Corollary (Handshaking lemma/degree sum formula)

Every graph has an even number of odd degree vertices.

Complete Graph

A graph G is complete if it is simple and all pairs of distnct vertices are adjacent. A complete graph on n vertices is denoted by K_n.



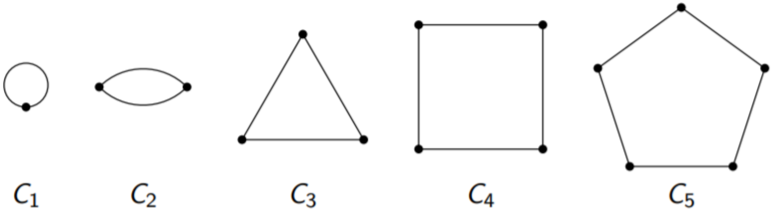
K_1 K_2 K_3 K_4 K_5

Cycle

A graph G is a cycle if $V(G)$ can be ordered as $v_1, \dots, v_n$, and $E(V)$ can be ordered as $e_1, \dots, e_n$, where

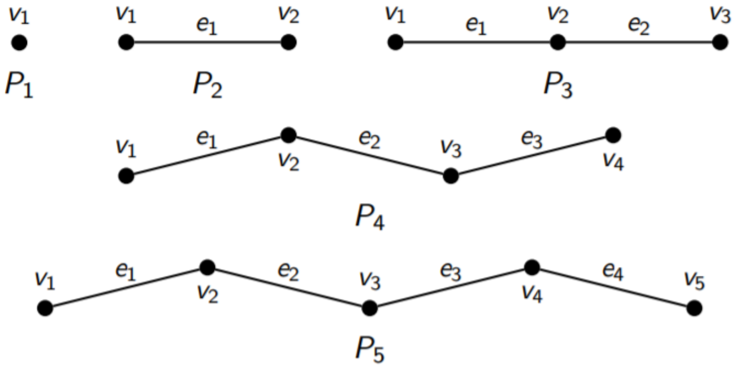
$$e_i = \begin{cases} v_i v_{i+1}, & 1 \le i \le n-1 \\ v_n v_1, & i = n \end{cases}$$

A cycle on n vertices is denoted by C_n.



Path

A graph G is called a path if the vertices can be ordered as $v_1, \ldots, v_n$, and edges can be ordered as $e_1, \ldots, e_{n-1}$ such that $e_i = v_i v_{i+1}, i = 1, \ldots, n$. A path on n vertices is denoted by P_n.



Walk

A walk W in a graph G is a sequence $v_0, e_1, v_1, \ldots, e_n, v_n$ such that every e_i has ends v_{i-1} and v_i. If $v_0 = v_n$, we say that W is closed. The length of a walk, path, or cycle is its number of edges. A walk is closed if its ends are the same.

Remark:

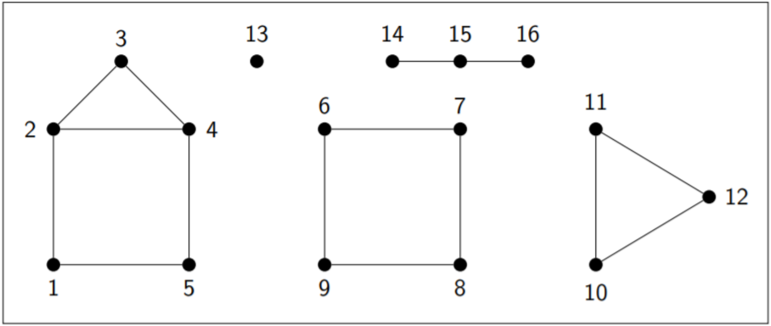
1. A walk is NOT a graph in general.
2. A path is a graph.
3. If $v_0, \ldots, v_n$ in a walk are distinct, we also call this walk a path.
4. A walk with only 1 vertex has length 0 .

Connected Graph

A graph G is connected if for all $u, v \in V(G)$, there is a walk from u to v (also called a u, v-walk). Otherwise, G is disconnected.

Component

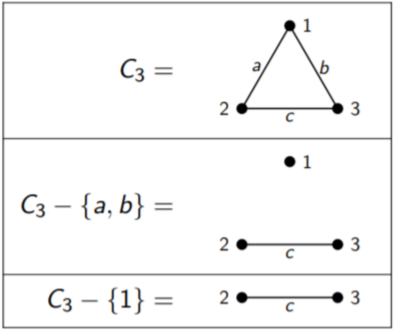
A component of a graph G is a maximal non-empty connected subgraph of G. The number of components of G is denoted $\mathrm{comp}(G)$.



$\mathrm{comp}(G) = 5$

Deleting

Given graph $G, S \subset E(G)$, then $G - S$ is the graph obtained from G by deleting S.
Given graph $G, X \subset V(G)$, then $G - X$ is the graph obtained from G by deleting every vertex in X and every edge incident to a vertex in X.

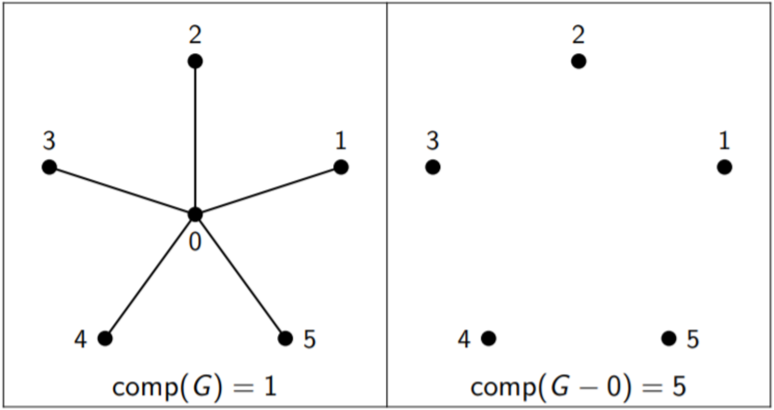


Bridge

An edge $e \in E(G)$ is called a cut-edge or bridge if no cycle contains e.
Given graph G and $e \in E(G)$, then
either e is a cut-edge and $\text{comp}(G - e) = \text{comp}(G) + 1$;
or e is NOT a cut-edge and $\text{comp}(G - e) = \text{comp}(G)$.

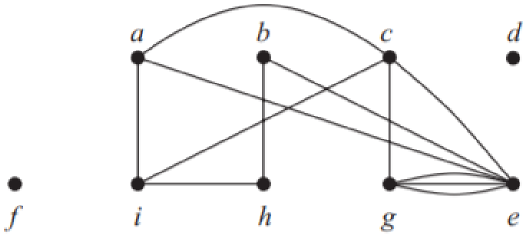
Cut-vertex

A vertex $v \in V(G)$ is called a cut-vertex whose deletion increases the number of components.



Exercise 1

Find the number of vertices, the number of edges, and the degree of each vertex in the given undirected graph. Identify all isolated and pendant vertices.



Exercise 1 Solution

There are 9 vertices here, and 12 edges. The degree of each vertex is the number of edges incident to it. Thus $\deg(a) = 3, \deg(b) = 2, \deg(c) = 4, \deg(d) = 0$ (and hence d is isolated), $\deg(e) = 6, \deg(f) = 0$ (and hence f is isolated), $\deg(g) = 4, \deg(h) = 2$, and $\deg(i) = 3$. Note that the sum of the degrees is $3 + 2 + 4 + 0 + 6 + 0 + 4 + 2 + 3 = 24$, which is twice the number of edges.

Definition

A bipartation of a graph G is a pair (A, B) where $A, B \subset V(G)$ with $A \cap B = \varnothing, A \cup B = V(G)$ such that every edge has an end in A and an end in $B.G$ is bipartite if it admits a bipartation.

Complete Bipartite Graph

A complete bipartite graph or biclique, denoted $K_{m,n}$, is a simple bipartite graph with bipartation (A, B) with $|A| = m$ and $|B| = n$ such that every vertex in A is adjacent to every vertex in B.

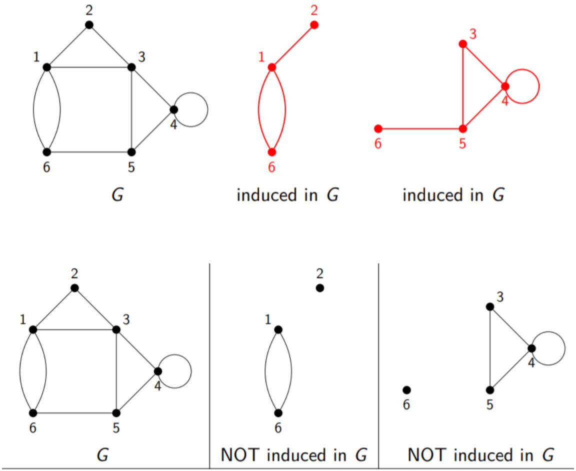
Induced Subgraph

A subgraph $H \subset G$ is induced if every edge of G with both ends in $V(H)$ is in $E(H)$. Equivalently, H is induced if $H = G - (V(G) \backslash V(H))$.

An induced path is sometimes called a snake.

An induced cycle is sometimes called a chordless cycle or a hole.

Induced Subgraph

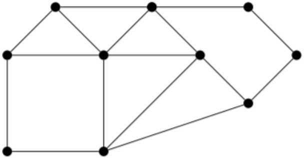


Matching

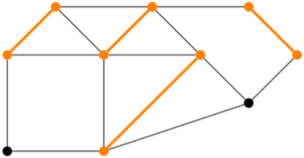
A matching in a graph $G = (V, E)$ is a subset of edges M such that M does not contain a loop and no two edges in M are incident with a common vertex. (i.e., the graph (V, M) has all vertices of degree < 2)

- A matching M is maximal if there is no matching M' such that $M \subsetneq M'$.
- A matching M is maximum if there is no matching M' such that $|M| < |M'|$.
- A perfect matching is a matching M such that every vertex of G is incident with an edge in M.

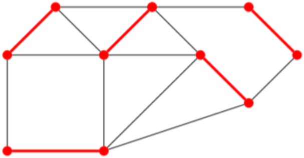
Matching



G



The matching M is maximal in G



The matching M' is maximal, maximum, and perfect in G

Hall's Theorem

Let G be a bipartite graph with bipartation (A, B). There exists a matching covering A iff there does not exist $X \subset A$ with $|N(X)| < |X|$.

Exercise 2

How many vertices and how many edges do these graphs have?

a) K_n
b) C_n
c) W_n
d) $K_{m,n}$
e) Q_n

Exercise 2 Solution

a) Obviously K_n has n vertices. It has $C(n,2) = n(n-1)/2$ edges, since each unordered pair of distinct vertices is an edge.
b) Obviously C_n has n vertices. Just as obviously it has n edges.
c) The wheel W_n is the same as C_n with an extra vertex and n extra edges incident to that vertex. Therefore it has $n+1$ vertices and $n+n=2n$ edges.
d) By definition $K_{m,n}$ has $m+n$ vertices. Since it has one edge for each choice of a vertex in the one part and a vertex in the other part, it has mn edges.
e) Since the vertices of Q_n are the bit strings of length n, there are 2^n vertices. Each vertex has degree n, since there are n strings that differ from any given string in exactly one bit (any one of the n different bits can be changed). Thus the sum of the degrees is $n2^n$. Since this must equal twice the number of edges (by the handshaking theorem), we know that there are $n2^n/2 = n2^{n-1}$ edges.

Exercise 3

A simple graph is called regular if every vertex of this graph has the same degree. A regular graph is called n-regular if every vertex in this graph has degree n.

For which values of n are these graphs regular?

a) K_n

b) C_n

c) W_n

d) Q_n

e) For which values of m and n is $K_{m,n}$ regular?

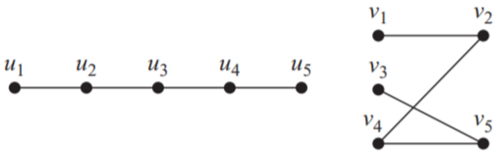
Exercise 3 Solution

a) The complete graph K_n is regular for all values of $n \geq 1$, since the degree of each vertex is $n-1$.
b) The degree of each vertex of C_n is 2 for all n for which C_n is defined, namely $n \geq 3$, so C_n is regular for all these values of n.
c) The degree of the middle vertex of the wheel W_n is n, and the degree of the vertices on the "rim" is 3 . Therefore W_n is regular if and only if $n = 3$. Of course W_3 is the same as K_4.
d) The cube Q_n is regular for all values of $n \geq 0$, since the degree of each vertex in Q_n is n. (Note that Q_0 is the graph with 1 vertex.)
e) Since the vertices in one part have degree m, and vertices in the other part have degree n, we conclude that $K_{m,n}$ is regular if and only if $m = n$.

Exercise 4

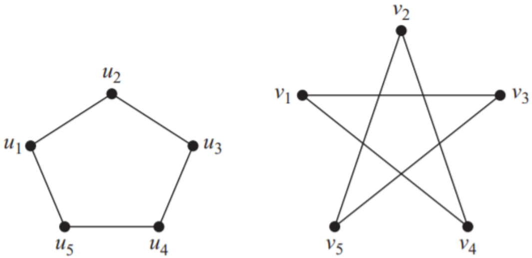
Determine whether the given pair of graphs is isomorphic. Exhibit an isomorphism or provide a rigorous argument that none exists.

a)



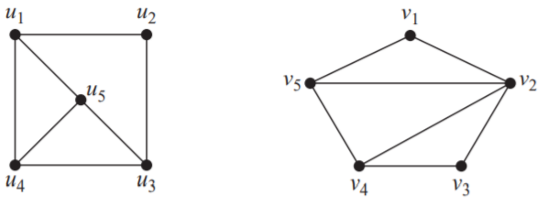
Exercise 4

b)



Exercise 4

c)



Exercise 4 Solution

a) These graphs are isomorphic, since each is a path with five vertices. One isomorphism is $f(u_1) = v_1, f(u_2) = v_2, f(u_3) = v_4, f(u_4) = v_5$, and $f(u_5) = v_3$.
b) These graphs are isomorphic, since each is the 5-cycle. One isomorphism is $f(u_1) = v_1, f(u_2) = v_3, f(u_3) = v_5, f(u_4) = v_2$, and $f(u_5) = v_4$.
c) These graphs are not isomorphic. The second has a vertex of degree 4, whereas the first does not.

Forest and Tree

Definition

A forest is a graph with no cycles. A tree is a connected forest.

Theorem

If G is is a forest, then comp $(G) = |V(G)| - |E(G)|$. In particular, if T is a tree, then $|V(T)| = |E(T)| + 1$.

Leaf

Definition

A leaf is a vertex of degree 1.

Theorem

Let T be a tree with $|V(T)| \geq 2$, then T has at least 2 leaves, and if there are only 2 leaves, then T is a path.

Corollary

If T is a tree and v is a leaf, then $T - v$ is a tree.

Spanning Tree

Definition

If T is a subgraph of a graph G, and T is a tree with $V(T) = V(G)$, then we call T a spanning tree of G.

Theorem

Let G be a connected graph with $|V(G)| \geq 2$. If H is a subgraph satisfying
(i) either H is minimal such that $V(H) = V(G)$ and H is connected,
(ii) or H is maximal such that H has no cycles,
then H is a spanning tree of G.

Kruskal's Algorithm

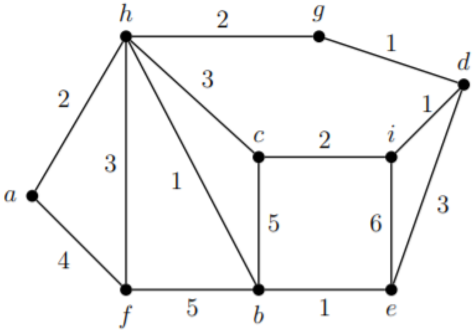
procedure Kruskal (G : weighted connected undirected graph with n vertices)
$T :=$ empty graph
for $i := 1$ to $n - 1$
  $e :=$ any edge in G with smallest weight that does not form a simple circuit when added to T
  $T := T$ with e added
return $T\{T$ is a minimum spanning tree of $G\}$

Kruskal's Algorithm

procedure Kruskal (G : weighted connected undirected graph with n vertices)
$T :=$ empty graph
for $i := 1$ to $n - 1$
  $e :=$ any edge in G with smallest weight that does not form a simple circuit when added to T
  $T := T$ with e added
return $T \{ T$ is a minimum spanning tree of $G \}$

Kruskal's Algorithm

Find a minimum-weight spanning tree via Kruskal's algorithm. List the edges chosen in order and sketch the tree.



Kruskal's Algorithm

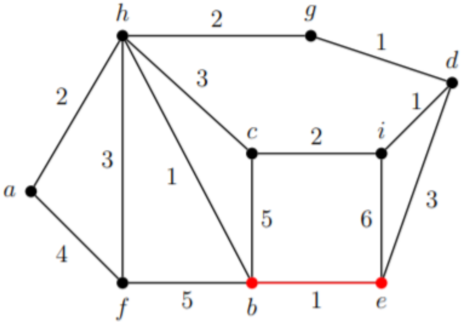
For Kruskal's algorithm, we first list the edges in the increasing order of its weighted length, which is given by

$$1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 5, 5, 6$$

The ties don't matter for the result(or one can say for different order of tied edges will give a specific minimum-weight spanning tree). We randomly choose the following order of edge and list as $be, bh, id, dg, ah, hg, ci, hf, hc, ed, af, fb, bc, ie$.

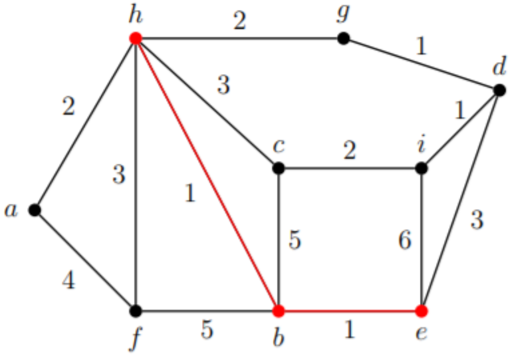
Kruskal's Algorithm

Now, we draw the following spanning tree by Kruskal's algorithm. First, we add *be* in our graph



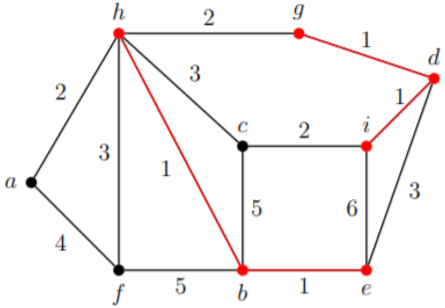
Kruskal's Algorithm

Next, we see that adding *bh* will not introduce a cycle, hence we add it in our graph



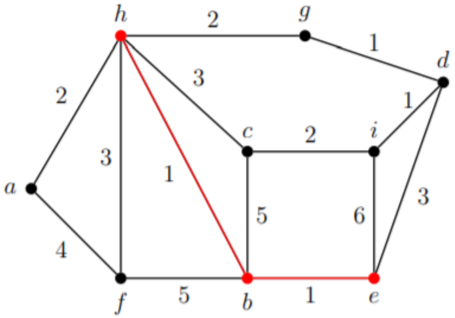
Kruskal's Algorithm

Next, we see that adding *id* will not introduce a cycle, hence we add it in our graph



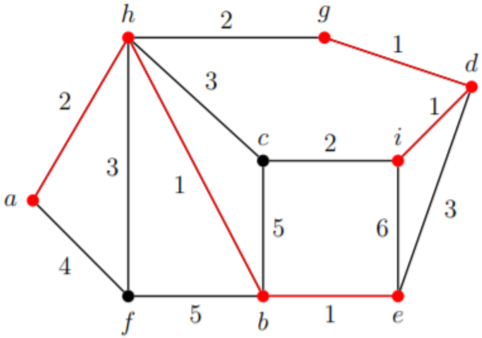
Kruskal's Algorithm

Next, we see that adding *dg* will not introduce a cycle, hence we add it in our graph



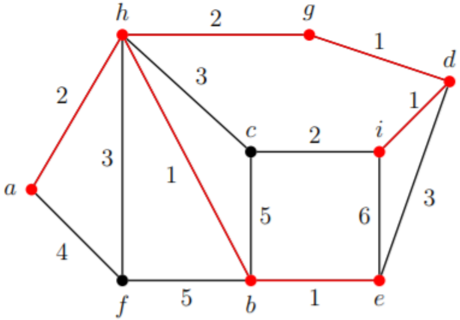
Kruskal's Algorithm

Next, we see that adding *ah* will not introduce a cycle, hence we add it in our graph



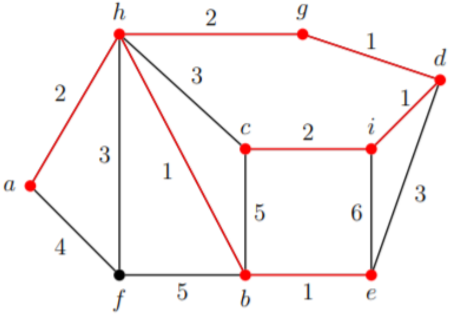
Kruskal's Algorithm

Next, we see that adding *hg* will not introduce a cycle, hence we add it in our graph



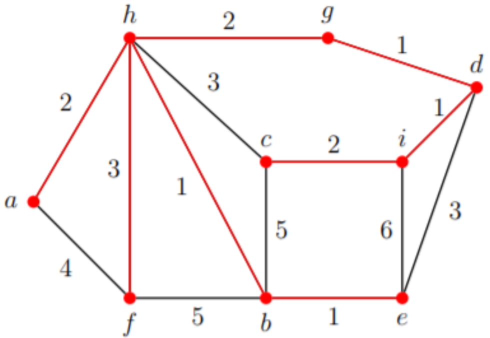
Kruskal's Algorithm

Next, we see that adding *ci* will not introduce a cycle, hence we add it in our graph



Kruskal's Algorithm

Next, we see that adding *hf* will not introduce a cycle, hence we add it in our graph



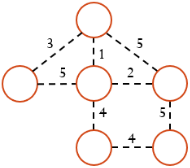
Kruskal's Algorithm

Next, we see that adding hc will introduce a cycle, hence we omit it. Also, adding ed, af, fb, bc or ie will all introduce a cycle, hence we terminate the algorithm, resulting a desired minimum-weight spanning tree.

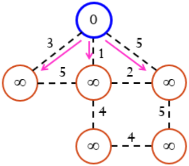
Prim's Algorithm

procedure Prim (G : weighted connected undirected graph with n vertices) $T :=$ a minimum-weight edge
for $i := 1$ to $n - 2$
  $e :=$ an edge of minimum weight incident to a vertex in T and not forming a simple circuit in T if added to T
  $T := T$ with e added
return $T\{T$ is a minimum spanning tree of $G\}$

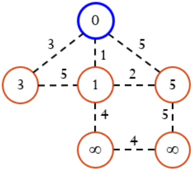
Prim's Algorithm



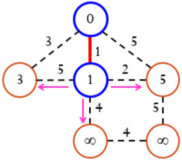
(a)



(b)

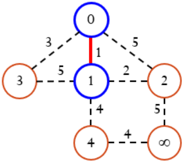


(c)

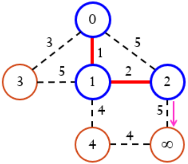


(d)

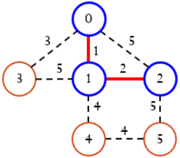
Prim's Algorithm



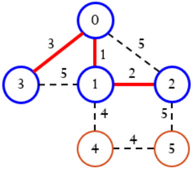
(e)



(f)

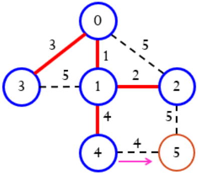


(g)

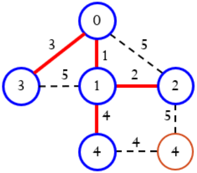


(h)

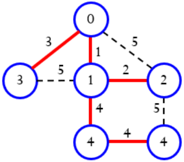
Prim's Algorithm



(i)



(j)



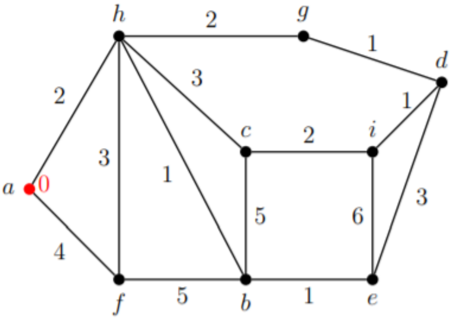
(k)

Dijkstra's Algorithm

Input: A simple connected graph $G = (V, E)$ with root vertex r and nonnegative weight function $w : E(G) \to \mathbb{R}_{\geq 0}$.
Output: A shortest path spanning tree for r.
- Procedure:
1. $i = 1$. Set T_1 to be the tree consisting of only the root vertex r.
2. $i \geq 2$. Choose an edge uv such that $u \in V(T_{i-1})$, $v \in V(G) \backslash V(T_{i-1})$, and dist $\mathrm{di}_T(r, u) + w(uv)$ is minimum. Let $T_i := T_{i-1} + uv$. If no such choice is possible, return the present tree.

Dijkstra's Algorithm

Given the root vertex *a*, find a shortest-path spanning tree via Dijkstra's algorithm. List the edges chosen in order, list the shortest path distance (from root vertex) to each vertex. Sketch the tree.

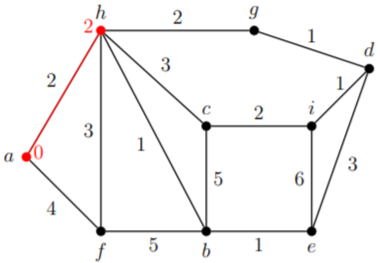


Dijkstra's Algorithm

We first find the neighbors of a and the temporary cost of them. The neighbors are h and f, with the temporary cost

$$\text{Cost}_{a \to h} = 2,$$
$$\text{Cost}_{a \to f} = 4.$$

We choose the minimal costed edge, which is ah and add it in the graph where we specify the determined cost from the root a to the node beside it in red.

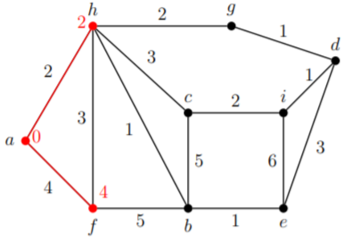


Dijkstra's Algorithm

We first find the neighbors of a, h and the temporary cost of them. The neighbors are f, g and c, with the temporary cost

$$\text{Cost}_{a\to f} = 4,$$
$$\text{Cost}_{h\to c} = 2 + 3 = 5,$$
$$\text{Cost}_{h\to g} = 2 + 2 = 4.$$

There is a tie, and we randomly choose one of which. In this case, we choose af to be added in the graph

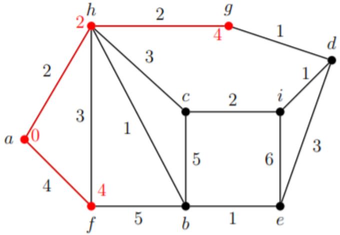


Dijkstra's Algorithm

Next, we consider the neighbors of h, f and the temporary cost of them. The neighbors are c, b and g, with the temporary cost

$$\text{Cost }_{h \to c} = 2 + 3 = 5,$$
$$\text{Cost }_{h \to g} = 2 + 2 = 4,$$
$$\text{Cost }_{f \to b} = 4 + 5 = 9.$$

We choose the minimal costed edge, which is hg and add it in the graph

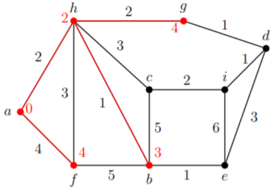


Dijkstra's Algorithm

Next, we consider the neighbors of h, f and g and the temporary cost of them. The neighbors are b, c and d, with the temporary cost

$$\begin{aligned} \text{Cost}_{h\to b} &= 2+1=3, \\ \text{Cost}_{f\to b} &= 4+5=9, \\ \text{Cost}_{h\to c} &= 2+3=5, \\ \text{Cost}_{g\to d} &= 4+1=5. \end{aligned}$$

We choose the minimal costed edge, which is hb and add it in the graph

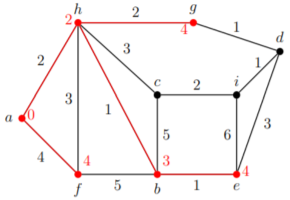


Dijkstra's Algorithm

Next, we consider the neighbors of h, b and g and the temporary cost of them. The neighbors are c, e and d, with the temporary cost

$$\text{Cost}_{h \to c} = 2 + 3 = 5$$
$$\text{Cost}_{b \to c} = 3 + 5 = 8$$
$$\text{Cost}_{b \to e} = 3 + 1 = 4$$
$$\text{Cost}_{g \to d} = 4 + 1 = 5.$$

Notice that we do not need to consider f anymore, since any edges expanded from which will introduce a cycle. We choose the minimal costed edge, which is be and add it in the graph

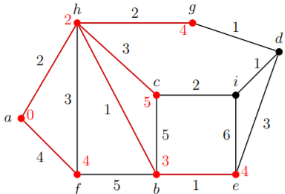


Dijkstra's Algorithm

Next, we consider the neighbors of h, b, e and g and the temporary cost of them. The neighbors are c, i and d, with the temporary cost

$$\text{Cost }_{h\to c} = 2+3 = 5$$
$$\text{Cost }_{b\to c} = 3+5 = 8$$
$$\text{Cost }_{e\to i} = 4+6 = 10$$
$$\text{Cost }_{g\to d} = 4+1 = 5$$
$$\text{Cost }_{e\to d} = 4+3 = 7$$

There is a tie, and we randomly choose one of which. In this case, we choose hc to be added in the graph

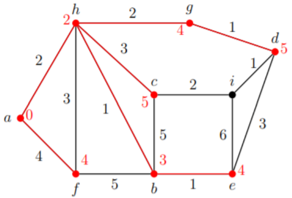


Dijkstra's Algorithm

Next, we consider the neighbors of c, e and g and the temporary cost of them. The neighbors are i and d, with the temporary cost

$$\begin{aligned}\text{Cost}_{c\to i} &= 5+2=7,\\ \text{Cost}_{e\to i} &= 4+6=10,\\ \text{Cost}_{g\to d} &= 4+1=5,\\ \text{Cost}_{e\to d} &= 4+3=7.\end{aligned}$$

Notice that we do not need to consider b anymore, since any edges expanded from which will introduce a cycle. We choose the minimal costed edge, which is gd and add it in the graph

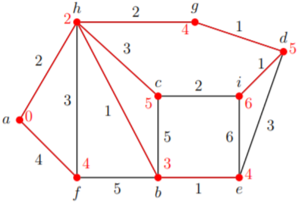


Dijkstra's Algorithm

Finally, we consider the neighbors of c, e and d and the temporary cost of them. The only neighbor left is i, with the temporary cost

$$\text{Cost}_{c\to i} = 5 + 2 = 7$$
$$\text{Cost}_{d\to i} = 5 + 1 = 6,$$
$$\text{Cost}_{e\to i} = 4 + 3 = 7$$

We choose the minimal costed edge, which is di and add it in the graph



Follow the procedure, one can see the order to choose the vertex adding in the graph, and also the red value labeled beside each

1. Graph
   - Graph Definition
   - Special Graphs
   - Connectivity
   - Bipartation
2. Tree
   - Tree Definition
   - Kruskal's Algorithm
   - Prim's Algorithm (Optional)
   - Dijkstra's Algorithm
3. **Combinatorial mathematics**
   - Choosing
   - Inclusion-Exclusion Principle
4. Q&A

Twelvefold Way

Distribute k balls into n urns. $(f : B \to U, |B| = k, |U| = n)$

Balls (domain)	Urns (codomain)	unrestricted (any function)	≤ 1 (injective)	≥ 1 (surjective)
labeled	labeled	n^k	$n^{\underline{k}}$	$n!\left\{ \begin{matrix} k \\ n \end{matrix} \right\}$
unlabeled	labeled	$\left(\!\!\binom{n}{k}\!\!\right)$	$\binom{n}{k}$	$\left(\!\!\binom{n}{k-n}\!\!\right)$
labeled	unlabeled	$\sum_{i=1}^{n} \left\{ \begin{matrix} k \\ i \end{matrix} \right\}$	$[k \leq n]$	$\left\{ \begin{matrix} k \\ n \end{matrix} \right\}$
unlabeled	unlabeled	$\sum_{i=1}^{n} p_i(k)$	$[k \leq n]$	$p_n(k)$

Twelvefold Way

$$n^{\underline{k}} = (n)_k = P(n,k) = P_k^n$$

$$\binom{n}{k} = C(n,k) = C_k^n$$

$$\left(\!\!\binom{n}{k}\!\!\right) = \binom{n+k-1}{k}$$

$\left\{ {k \atop n} \right\} = \#$ partition of $[k]$ into n parts.

$p_n(k) = \#$ partition of k into n parts.

$[k \leq n]$: Iverson bracket

Exercise 5

Consider the functions $f : B \to U$, count the number of functions and fill in the blanks below. Express the results in binomial coefficients, factorials, or powers (AVOID double bracket notation for first two rows).

Domain	Codomain	Any	Injective	Surjective
distinguishable	distinguishable			
indistinguishable	distinguishable			
distinguishable	indistinguishable			
indistinguishable	indistinguishable			

where

(i) $B = \{1, 2, 3\}$ and $U = \{1, 2, 3, 4, 5\}$.

(ii) $B = \{1, 2, 3, 4, 5\}$ and $U = \{1, 2, 3\}$.

Exercise 5 Solution

For

$$f : B \to U, \quad B = \{1, 2, 3\}, U = \{1, 2, 3, 4, 5\}$$

we have

Domain	Codomin	Any	Injectve	surjective
distinguishable	distinguishable	5^3	$\binom{5}{3} \cdot 3!$	0
indistinguishable	distinguishable	$\binom{7}{3}$	$\binom{5}{3}$	0
distinguishable	indistinguishable	5	1	0
indistinguishable	indistinguishable	3	1	0

Exercise 5 Solution

For

$$f : B \to U, \quad B = \{1, 2, 3, 4, 5\}, U = \{1, 2, 3\}$$

we have

Domain	Codomain	Any	Injective	Surjective
distinguishable	distinguishable	3^5	0	$30 \times 3!$
indistinguishable	distinguishable	$\binom{7}{5}$	0	$\binom{4}{2}$
distinguishable	indistinguishable	41	0	25
indistinguishable	indistinguishable	5	0	2

Stirling numbers of the second kind

$$\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = S(n,k) = S_n^{(k)}$$

$$= \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \left(\begin{matrix} k \\ i \end{matrix} \right) (k-i)^n$$

k	0	1	2	3	4	5	6
0	1						
1	0	1					
2	0	1	1				
3	0	1	3	1			
4	0	1	7	6	1		
5	0	1	15	25	10	1	
6	0	1	31	90	65	15	1

Multichoosing

Let $\left(\!\left(\begin{array}{c} n \\ k \end{array}\right)\!\right)$ be the number of k-element multisets on an n-element set. Reads " n multichoose k ".

$$\left(\!\left(\begin{array}{c} n \\ k \end{array}\right)\!\right) = \left(\begin{array}{c} n+k-1 \\ k \end{array}\right) = \left(\begin{array}{c} n+k-1 \\ n-1 \end{array}\right)$$

Exercise 6

Consider

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \leq 100$$

What are the number of integer solutions if
(i) $x_i > 0$ and $=$ holds;
(ii) $x_i \geq 0$ and $=$ holds;
(iii) $x_i > 0$ and $<$ holds;
(iv) $x_i \geq 0$ and $<$ holds;
(v) $x_i \geq 0$.
AVOID double bracket notation in the final solution.

Exercise 6 Solution

i) If $x_i > 0$ and the equality holds. Namely, we have

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 100$$

with $x_i > 0$, or equivalently, $x_i \geq 1$. Then we can divide 100 into

$$\underbrace{1 + 1 + \cdots + 1}_{100 \text{terms}} = 100$$

Since we have 7 variables, then the number of integer solutions is given by

$$\binom{99}{6}$$

which stands for choosing 6 of $+$ sign out of $99+$ sign to form a $7-$ partition of 100 .

Exercise 6 Solution

ii) If $x_i \geq 0$ and the equality holds. Namely, we have

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 100$$

with $x_i \geq 0$, or equivalently, we have

$$x_1' + x_2' + x_3' + x_4' + x_5' + x_6' + x_7' = 107$$

with $x_i' > 0$, or equivalently, $x_i' \geq 1$ since $x_i' := x_i + 1$. Then by the same argument from (i), we know that the number of integer solutions is given by

$$\binom{106}{6}$$

Exercise 6 Solution

iii) If $x_i > 0$ and the equality does not hold. We introduce another variable $x_8 > 0$ such that

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 100$$

which is equivalent to the original one. Then by the same argument from (i), we know that the number of integer solutions is given by

$$\binom{99}{7}$$

Exercise 6 Solution

iv) If $x_i \geq 0$ and the equality does not hold. We introduce another variable $x_8 > 0$ such that

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 100,$$

which is equivalent to the original one. By the same transformation of variables as in (ii), we have

$$x_1' + x_2' + x_3' + x_4' + x_5' + x_6' + x_7' + x_8 = 107,$$

then now all the variables are strictly greater than 0 . By the same argument as in (i), we know that the number of integer solution is given by

$$\binom{106}{7}$$

Exercise 6 Solution

v) If $x_i \geq 0$. Then the number of integer solution is given by the results in (ii) and (iv) combined. Since we know that the solution sets in the case of

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 100$$

and the case of

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 < 100$$

are mutually disjoint, hence the number of solution can be combined, namely

$$\binom{106}{6} + \binom{106}{7} = \binom{107}{7}$$

by the recursive identity for binomial coefficient.

Counting Integer Solutions

How many nonnegative integer solutions to the equation

$$x_1 + x_2 + x_3 + x_4 = 63$$

such that $x_1, x_2 \geq 0, 2 \leq x_3 \leq 5, x_4 > 0$.
Consider the following solution sets, where $A \supset B$.
Answer: $|A| - |B|$.
A : such that $x_1, x_2 \geq 0, x_3 \geq 2, x_4 > 0$, i.e.,
$x_3 - 2 \geq 0, x_4 - 1 \geq 0$, and

$$x_1 + x_2 + (x_3 - 2) + (x_4 - 1) = 60$$

We have $|A| = \binom{60+3}{3}$.
B : such that $x_1, x_2 \geq 0, x_3 > 5, x_4 > 0$, i.e.,
$x_3 - 6 \geq 0, x_4 - 1 \geq 0$, and

$$x_1 + x_2 + (x_3 - 6) + (x_4 - 1) = 56$$

We have $|B| = \binom{56+3}{3}$.

Inclusion-Exclusion Principle

2 sets: $|A \cup B| = |A| + |B| - |A \cap B|$

3 sets: $|A \cup B \cup C| = |A| + |B| + |C|$

$-|A \cap B| - |B \cap C| - |C \cap A| + |A \cap B| - |B \cap C|$

Inclusion-Exclusion Principle

Let $A_1, \ldots, A_n$ be subsets of X. Then the number of elements of X which lie in none of the subsets A_i is

$$\sum_{I \subset \{1,\ldots,n\}} (-1)^{|I|} |A_I|$$

Exercise 7

Find the number of solutions of the equation $x_1 + x_2 + x_3 + x_4 = 17$, where $x_i, i = 1, 2, 3, 4$, are nonnegative integers such that $x_1 \leq 3, x_2 \leq 4, x_3 \leq 5$, and $x_4 \leq 8$.

Exercise 7 Solution

$$C(4+17-1,17)-C(4+13-1,13)-C(4+12-1,12)-C(4+11-1,11)-C(4+8-1,8)+C(4+8-1,8)+C(4+7-1,7)+C(4+4-1,4)+C(4+6-1,6)+C(4+3-1,3)+C(4+2-1,2)-C(4+2-1,2)=20$$

Derangement

$$D_n = \sum_{i=0}^{n} (-1)^i \binom{n}{i} (n-i)! = n! \sum_{i=0}^{n} \frac{(-1)^i}{i!}$$

Exercise 8

What is the probability that none of 10 people receives the correct hat if a hatcheck person hands their hats back randomly?

Exercise 8 Solution

$$\frac{D_{10}}{10!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{1}{10!} = \frac{1334961}{3628800} = \frac{16481}{44800} \approx 0.3678794643,$$

which is almost exactly $e^{-1} \approx 0.3678794412\ldots$

Exercise 9

A machine that inserts letters into envelopes goes haywire and inserts letters randomly into envelopes. What is the probability that in a group of 100 letters
a) no letter is put into the correct envelope?
b) exactly one letter is put into the correct envelope?
c) exactly 98 letters are put into the correct envelopes?
d) exactly 99 letters are put into the correct envelopes?
e) all letters are put into the correct envelopes?

Exercise 9 Solution

a) An arrangement in which no letter is put into the correct envelope is a derangement. There are by definition D_{100} derangements. Since there are $P(100, 100) = 100\ !$ equally likely permutations altogether, the probability of a derangement is $D_{100}/100\ !$. Numerically, this is almost exactly equal to $1/e$, which is about 0.368.

b) We need to count the number of ways to put exactly one letter into the correct envelope. First, there are $C(100, 1) = 100$ ways to choose the letter that is to be correctly stuffed. Then there are D_{99} ways to insert the remaining 99 letters so that none of them go into their correct envelopes. By the product rule, there are $100D_{99}$ such arrangements. As in part (a) the denoninator is $P(100, 100) = 100\ !$. Therefore the answer is $100D_{99}/100! = D_{99}/99!$. Again this is almost exactly $1/e \approx 0.368$.

Exercise 9 Solution

c) This time, to count the number of ways that exactly 98 letters can be put into their correct envelopes, we need simply to choose the two letters that are to be misplaced, since there is only one way to misplace them. There are of course $C(100, 2) = 4950$ ways to do this. As in part (a) the denominator is $P(100, 100) = 100\,!$. Therefore the answer is $4950/100\,!$. This is substantially less than 10^{-100}, so for all practical purposes, the answer is 0 .
d) There is no way that exactly 99 letters can be inserted into their correct envelopes, since as soon as 99 letters have been correctly inserted, there is only one envelope left for the remaining letter, and it is the correct one. Therefore the answer is exactly 0 . (The probability of an event that cannot happen is 0 .)
e) Only one of the $100\,!$ permutations is the correct stuffing, so the answer is $1/100!$. As in part (c) this is 0 for all practical purposes.

Q&A

Q&A