- **Definition 6.4: Let $G=<V,E>$, $G'=<V',E'>$** be two graphs (both undirected or both directed).

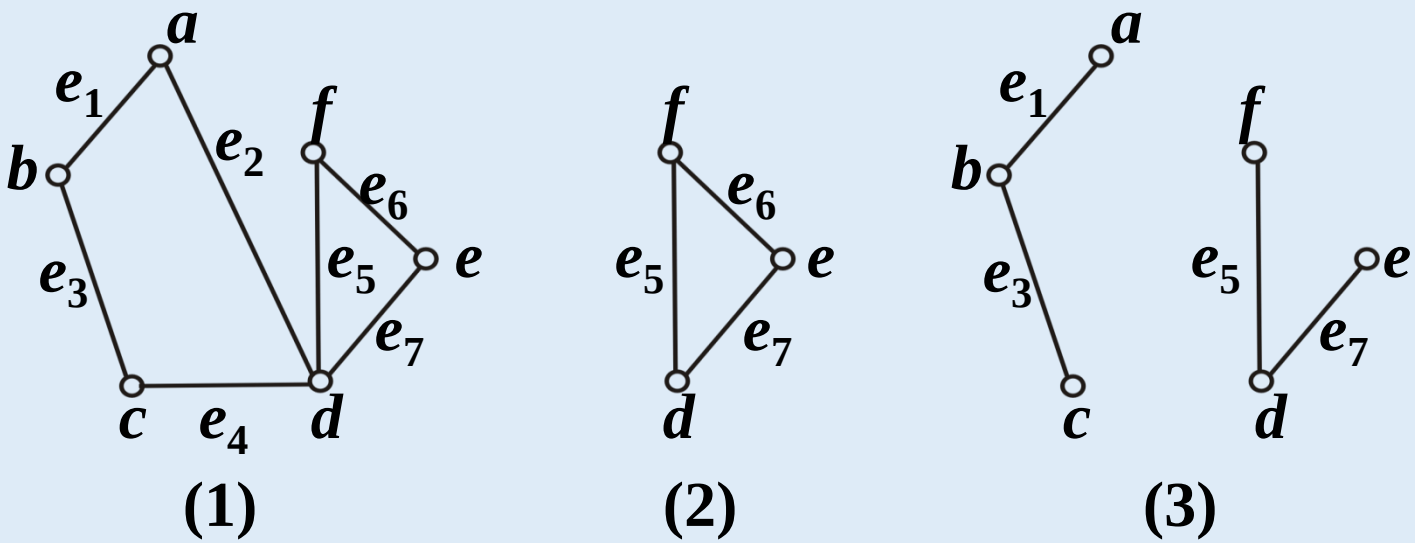
  (1) ***If $V'\subseteq V$* and $E'\subseteq E$, then G' is *G's subgraph*, G is called G's *supergraph*, denoted as $G'\subseteq G$.**

  (2) ***If $G'\subseteq G$ and $V'=V$*, then G' is called a *spanning subgraph* of G.**

  (3) ***If $V'\subset V$ or $E'\subset E$*, then G' is called a *proper subgraph* of G.**

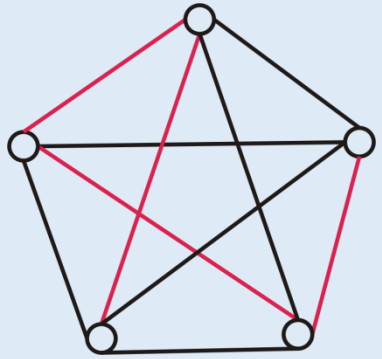
  (4) ***Let $V'\subseteq V$ and $V'\neq\varnothing$*, the subgraph of G with vertex set V' and edge set consisting of all edges in G whose endpoints are both in V' is called the *vertex-induced subgraph* on V' ,denoted by $G[V']$.**

  (5) ***Let $E'\subseteq E$ and $E'\neq\varnothing$*, the subgraph of G with edge set E' and vertex set consisting of all endpoints of the edges in E' is called the *edge-induced subgraph* on E' ,denoted by $G[E']$.**



- **(1),(2),(3) is (1)** subgraphs , **(2),(3)** are proper subgraphs, **(1)** is the supergraph.
- **(1),(3)** are spanning subgraphs of **(1).**
- **(2)** is the vertex-induced subgraph on **{*d*,*e*,*f* }**, and also the edge-induced subgraph on **{e_5, e_6, e_7}.**
- **(3)** is the edge-induced subgraph on **{e_1, e_3, e_5, e_7}.**

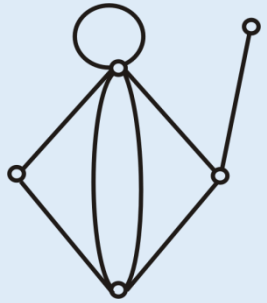
- **Definition 6.5:** Let $G=<V,E>$ be a simple undirected graph of order n, let $\overline{E} = V\&V - E$, then $\overline{G} =< V, E >$ is called the *complement* of graph G.

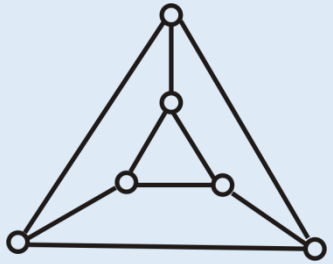


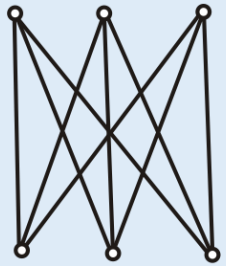
- 6.1.1 Undirected and Directed Graphs
- 6.1.2 Vertex Degree and the Handshaking Lemma
- 6.1.3 Common Types of Graphs
- 6.1.4 Subgraphs and Complements
- **6.1.5 Graph Isomorphism**

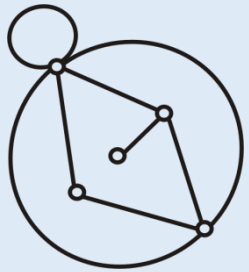
- **Definition 6.6:** Let $G_1=<V_1,E_1>$, $G_2=<V_2,E_2>$ be two undirected graphs (or two directed graphs), if there exists a bijective function f: $V_1 \rightarrow V_2$, such that for any $v_i,v_j \in V_1$:
  - $(v_i,v_j) \in E_1$ if and only if $<f(v_i), f(v_j)> \in E_2$
  - and the multiplicity of (v_i,v_j) **is equal to** $(f(v_i),f(v_j))$
  - then G_1 **and** G_2 are said to be *isomorphic*, denoted by $G_1 \cong G_2$.

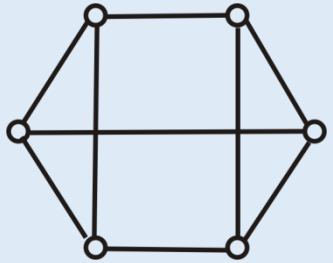
↳ Isomorphic graphs（e.g.）

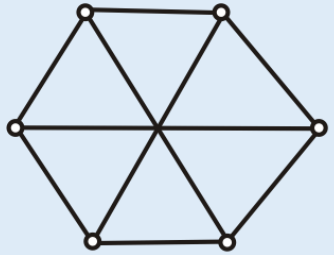




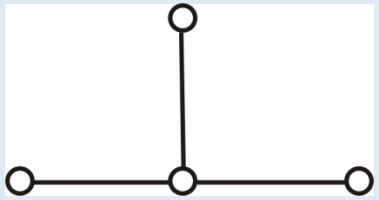




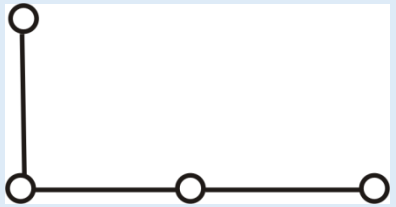




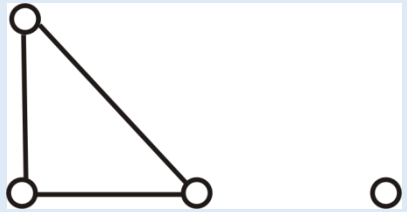
- **Example :** Draw all non-isomorphic simple undirected graphs with **4** vertices and **3** edges.
- **Solution:**
  - The total degree is **6** (since the sum of degrees equals twice the number of edges). This must be distributed among **4** vertices.
  - The maximum degree is **3**, and the number of vertices with odd degree must be even.
  - There are three possible degree sequences:**1, 1, 1, 3**； **1, 1, 2, 2**； **0, 2, 2, 2**



1,1,1,3

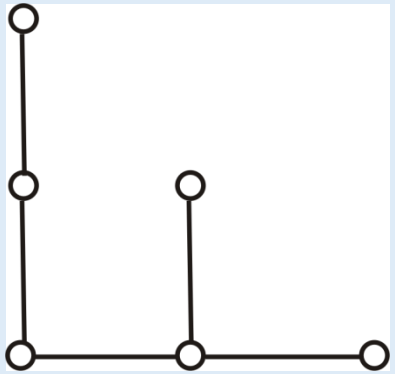


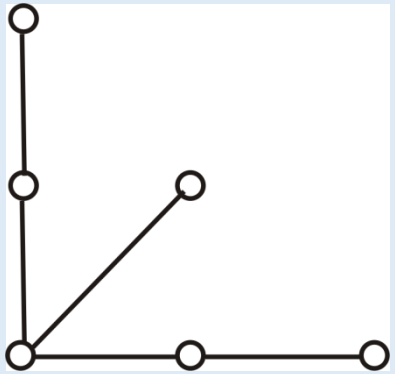
1,1,2,2

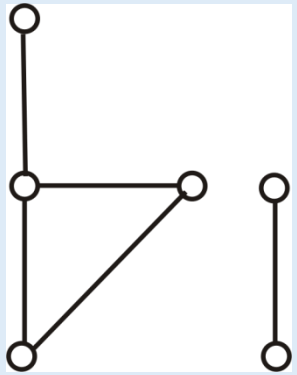


0,2,2,2

- **Example**: Draw three non-isomorphic simple undirected graphs with the degree sequence **1,1,1,2,2,3.**







Objective :

Key Concepts :

Discrete Mathematics 2025 Spring

同济经管 TONGJI SEM

魏可佶 kejiwei@tongji.edu.cn

CAMEA 中国高质量MBA教育认证 AACSB ACCREDITED EQUIS ACCREDITED

- 6.1 Basic Concepts of Graphs
- 6.2 Graph Connectivity
- 6.3 Matrix Representations of Graphs
- 6.4 Special Types of Graphs

- 6.2.1 Paths and Circuits

  Elementary Paths (Circuits) and Simple Paths (Circuits)

- **6.2.2 Connectivity and Connectedness in Undirected Graphs**

  Connected Graphs and Connected Components

  Shortest Paths and Distances

  Vertex Cut Sets, Cut Vertices, Edge Cut Sets, and Bridges

  Vertex Connectivity and Edge Connectivity

- **6.2.3 Connectivity and Classification in Directed Graphs**

  Reachability

  Weak Connectivity, Unilateral Connectivity, and Strong Connectivity

  Shortest Paths and Distances

- **Definition 6.8:** Given a graph $G=<V,E>$(either undirected or directed), an alternating sequence of vertices and edges in G $\Gamma=v_0e_1v_1e_2\ldots e_lv_l$.

  (1) $\forall i(1\le i\le l)$, $e_i=(v_{i-1},v_i)$. Γ is called a ***path*** from v_0 **to** v_l, v_0 **and** v_l are called the **starting point** and **ending point** of the path, respectively, and l is the **length** of the path. **If** $v_0=v_l$, Γ the path is called a ***circuit*** (or **cycle**).

  (2) If all the vertices in a path or circuit are distinct (except for $v_0=v_l$ in the case of a circuit), it is called an ***elementary path*** or **simple path** (and an **elementary circuit** or **simple cycle**). A cycle of odd length is called an **odd cycle**, and a cycle of even length is called an **even cycle**.

  (3) If all edges in a path or circuit are distinct, it is called a ***simple path*** (or **simple circuit**); otherwise, it is called a **non-simple** or **complex path** (or **complex circuit**).

Representations of a Path or Circuit

① According to the definition, using an ***alternating sequence*** of vertices and edges: $\Gamma=v_0e_1v_1e_2\ldots e_lv_l$.

② Using a ***sequence of edges***: $\Gamma=e_1e_2\ldots e_l$.

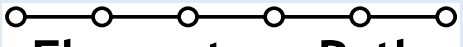
③ In a simple graph, using a ***sequence of vertices***: $\Gamma=v_0v_1\ldots v_l$

Properties of Circuits

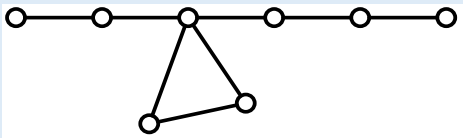
- In an **undirected graph**, a ***circuit of length 1*** is formed by a ***loop*** (an edge connecting a vertex to itself). A ***circuit of length 2*** is formed by *two parallel edges* between the same pair of vertices. In an **undirected simple graph**, all ***circuits have length ≥ 3***.
- In a **directed graph**, a circuit of length 1 is also formed by a **loop**. In a **directed simple graph**, all circuits have length ≥ 2.

↳ Elementary paths (or circuit) ⊂ simple paths (or circuit)

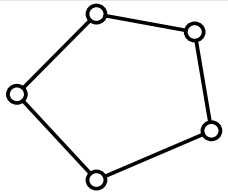
- Every elementary path (or circuit) is a simple path (or circuit), but the converse is not necessarily true.
- **Example:**



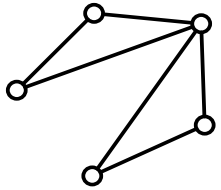
Elementary Path



Non-elementary simple path



Elementary Cycle



Non-elementary simple cycle

- 6.2.1 Paths and Circuits
  - Elementary Paths (Circuits) and Simple Paths (Circuits)
- 6.2.2 Connectivity and Connectedness in Undirected Graphs
  - Connected Graphs and Connected Components
  - Shortest Paths and Distances
  - Vertex Cut Sets, Cut Vertices, Edge Cut Sets, and Bridges
  - Vertex Connectivity and Edge Connectivity
- **6.2.3 Connectivity and Classification in Directed Graphs**
  - Reachability
  - Weak Connectivity, Unilateral Connectivity, and Strong Connectivity
  - Shortest Paths and Distances

- Let $G=<V,E>$ be an undirected graph, and let $u,v \in V$
  - ***Connectivity between u and v***: Vertex u is said to be **connected** to vertex v if there exists a path between them. By convention, every vertex is considered to be connected to itself.
  - ***Connected graph:*** A graph in which every pair of vertices is connected. A **trivial graph** (with only one vertex) is considered connected.
  - ***Connectivity relation***: $R=\{<u,v> \mid u,v \in V$ **and** u **is connected to** $v\}$. R is an **equivalence relation**.
  - ***Connected component***: The subgraph induced by each equivalence class of V under the relation R is called a **connected component** of G. Suppose $V/R=\{V_1,V_2,\ldots,V_k\}$, then the connected components of G are the subgraphs $G[V_1],G[V_2],\ldots,G[V_k]$.
  - ***Number of connected components***: $p(G)=k$

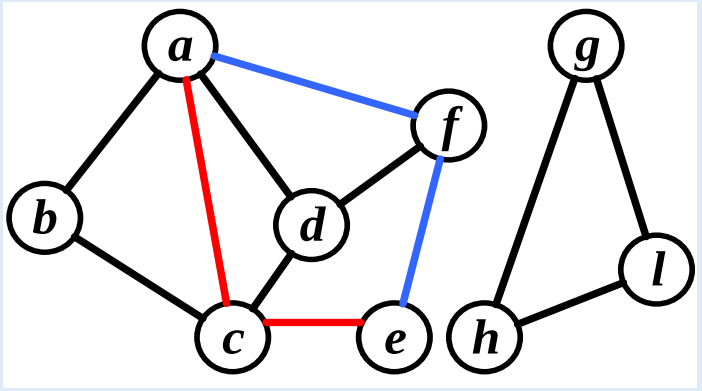
    G is a connected graph $\Leftrightarrow p(G)=1$.

- ***Shortest path between u and v*: A path of the shortest length between vertices *u and v*, assuming *u and v* are connected.**
- ***Distance between u and v d(u,v)*: The length of the shortest path between *u and v*. If *u and v* are not connected, define $d(u,v)=\infty$.**
- **Property:**

  (1) $d(u,v)\geq 0$, and $d(u,v)=0 \Leftrightarrow u=v$

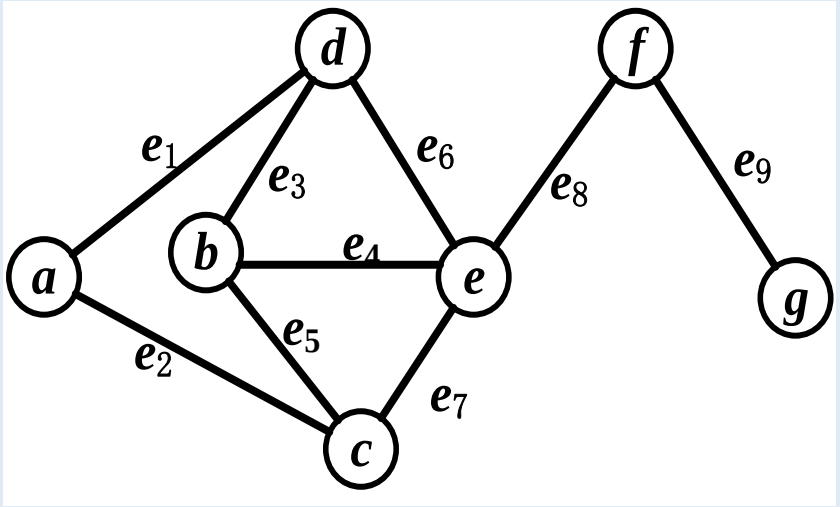
  (2) $d(u,v)=d(v,u)$

  (3) $d(u,v)+d(v,w)\geq d(u,w)$



- **For example:** The shortest path between ***a*** and ***e*** is shown in the figure on the right: ***ace***, ***afe***. $d(a,e)=2$, $d(a,h)=\infty$

- Let undirected graph $G=<V,E>$, $v \in V$, $e \in E$, $V' \subseteq V$, $E' \subseteq E$.
  - $G-v$: The graph obtained by **removing vertex v and all edges** incident to it from G.
  - $G-V'$: The graph obtained by **removing all vertices in V'** and their incident edges from G.
  - $G-e$: The graph obtained by **removing edge e** from G.
  - $G-E'$: The graph obtained by **removing all edges in E'** from G.
- **Definition 6.9:** Let undirected graph $G=<V,E>$, $V' \subset V$, If $p(G-V')>p(G)$, then V' is called a ***vertex cut set*** of G. if $\{v\}$ is vertex cut set, then v is called a ***cut vertex***.
  - Let $E' \subseteq E$, if $p(G-E')>p(G)$, then E' is called an ***edge cut set*** **of G.** If $\{e\}$ is edge cut set , then e s called a ***cut edge*** or a ***bridge***.



Cut vertex: e,f

Vertex cut set : $\{e\}$,$\{f\}$, $\{c,d\}$

Bridge: e_8, e_9

Edge cut set :$\{e_8\}$,$\{e_9\}$, $\{e_1,e_2\}$, $\{e_1, e_3, e_6\}$, $\{e_1, e_3, e_4, e_7\}$

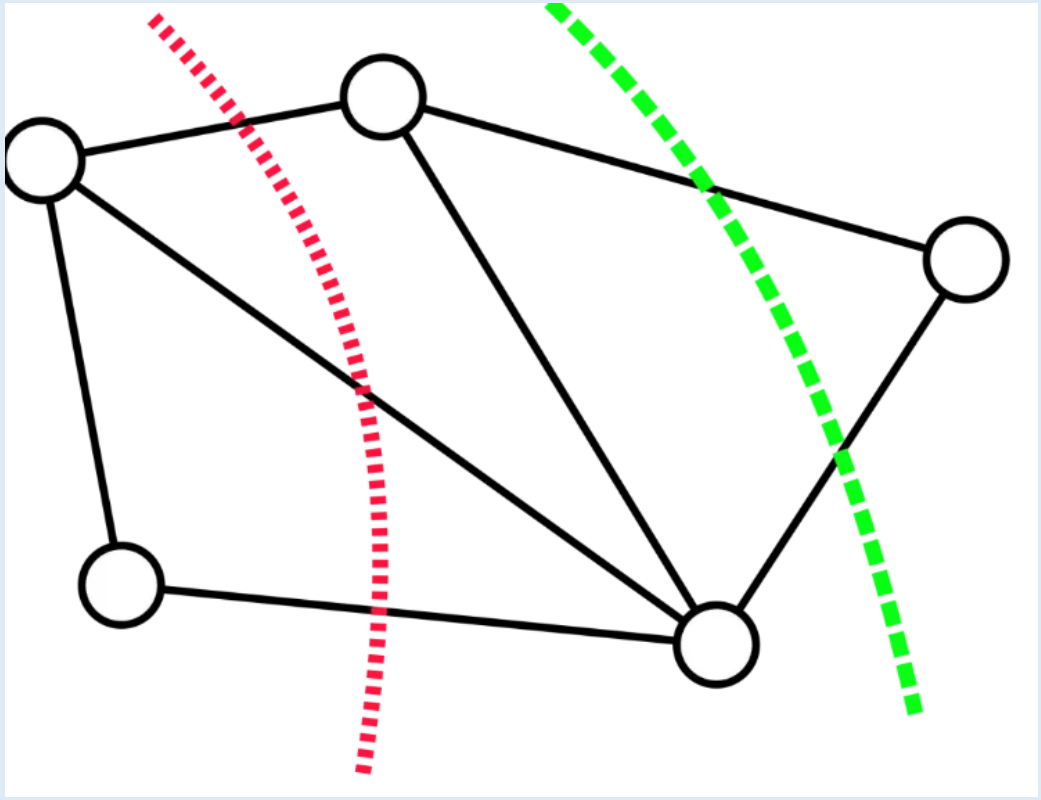
- **Notes:**

(1) The *complete graph* K_n has no vertex cut set.

(2) An *n-vertex null graph* has neither a vertex cut set nor an edge cut set.

(3) If G is connected and E' is an edge cut set, then $p(G-E')=2$.

(4) If G is connected and V' is a vertex cut set, then $p(G-V')\geq 2$.



Deterministic Near-Linear Time Minimum Cut in Weighted Graphs

Monika Henzinger* Jason Li† Satish Rao‡ Di Wang§

January 12, 2024

Abstract

In 1996, Karger [Kar96] gave a startling randomized algorithm that finds a minimum-cut in a (weighted) graph in time $O(m \log^3 n)$ which he termed near-linear time meaning linear (in the size of the input) times a polylogarthmic factor. In this paper, we give the first deterministic algorithm which runs in near-linear time for weighted graphs.

Previously, the breakthrough results of Kawarabayashi and Thorup [KT19] gave a near-linear time algorithm for simple graphs (which was improved to have running time $O(m \log^2 n \log\log n)$ in [HRW20].) The main technique here is a clustering procedure that perfectly preserves minimum cuts. Recently, Li [Li21] gave an $m^{1+o(1)}$ deterministic minimum-cut algorithm for weighted graphs; this form of running time has been termed "almost-linear". Li uses almost-linear time deterministic expander decompositions which do not perfectly preserve minimum cuts, but he can use these clusterings to, in a sense, "derandomize" the methods of Karger.

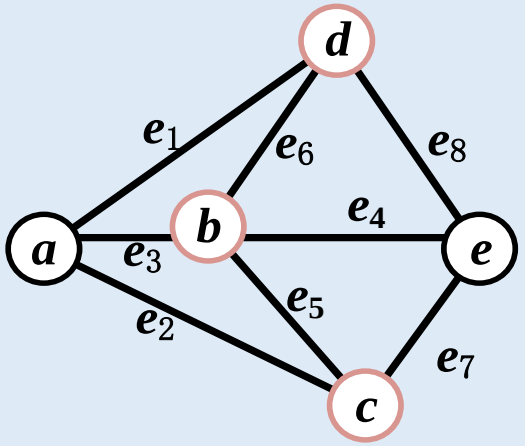
In terms of techniques, we provide a structural theorem that says there exists a sparse clustering that preserves minimum cuts in a weighted graph with $o(1)$ error. In addition, we construct it deterministically in near linear time. This was done exactly for simple graphs in [KT19, HRW20] and with polylogarithmic error for weighted graphs in [Li21]. Extending the techniques in [KT19, HRW20] to weighted graphs presents significant challenges, and moreover, the algorithm can only polylogarithmically approximately preserve minimum cuts. A remaining challenge is to reduce the polylogarithmic-approximate clusterings to $1+o(1/\log n)$-approximate so that they can be applied recursively as in [Li21] over $O(\log n)$ many levels. This is an additional challenge that requires building on properties of tree-packings in the presence of a wide range of edge weights to, for example, find sources for local flow computations which identify minimum cuts that cross clusters.

*Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria. See funding information in the acknowledgement section. email: monika.henzinger@ista.ac.at
†Simons Institute, UC Berkeley. email: jmli@alumni.cmu.edu
‡UC Berkeley. email: satishr@berkeley.edu
§Google Research. email: wadi@google.com

A graph and its ***two cuts***: the red dashed lines indicate a cut consisting of three edges, while the green lines represent a ***minimum cut of the graph***, consisting of two edges.

- **Definition 6.10:** An undirected connected graph $G=<V,E>$,
  - $\kappa(G)=\min\{|V'| \mid V'$ is a vertex cut of G *or* $G-V'$ becomes a trivial graph$\}$ is called the *vertex connectivity* of G.
  - $\lambda(G)=\min\{|E'| \mid E'$ is an edge cut of $G\}$ is called the *edge connectivity* of G.

- **Example:**



$\kappa(G)=3$

$\lambda(G)=3$

- **Notes:**
  - (1) If G is a *trivial graph*, then $\kappa(G)=0$, $\lambda(G)=0$.
  - (2) If G is a *complete graph* K_n, then $\kappa(G)=n-1$, $\lambda(G)=n-1$.
  - (3) If G has a *cut vertex*, then $\kappa(G)=1$; if G has a *bridge* (cut edge), then $\lambda(G)=1$.
  - (4) By convention, the vertex connectivity and edge connectivity of a *disconnected graph* are both defined to be 0.

- **Theorem 6.3:** For *any undirected graph* G, we have

$$\kappa(G) \leq \lambda(G) \leq \delta(G).$$

- 6.2.1 Paths and Circuits
  - Elementary Paths (Circuits) and Simple Paths (Circuits)
- 6.2.2 Connectivity and Connectedness in Undirected Graphs
  - Connected Graphs and Connected Components
  - Shortest Paths and Distances
  - Vertex Cut Sets, Cut Vertices, Edge Cut Sets, and Bridges
  - Vertex Connectivity and Edge Connectivity
- **6.2.3 Connectivity and Classification in Directed Graphs**
  - Reachability
  - Weak Connectivity, Unilateral Connectivity, and Strong Connectivity
  - Shortest Paths and Distances

- Let $D=<V,E>$ be a **directed graph**, $u,v \in V$,
  - (1) ***u* is** reachable from ***v*:** There exists a path from ***u*** to ***v***. By convention, every vertex is reachable from itself.
  - (2) ***u* and *v*** are ***mutually reachable***: ***u*** is reachable from ***v***, and ***v*** is reachable from ***u***.
  - (3) ***D* is *weakly connected* (connected)**: The undirected graph obtained by ignoring the directions of all edges is connected.
  - (4) ***D* is *unilaterally connected***: $\forall u,v \in V$, either ***u*** is reachable from ***v*** or v is reachable from ***u***.
  - (5) ***D* is *strongly connected***: $\forall u,v \in V$, ***u*** and ***v*** are mutually reachable.
  - (6) ***D* is *strongly connected*** if and only if there ***exists a circuit*** that passes through all vertices.
  - (7) ***D* is *unilaterally connected*** if and only if there ***exists a path*** that passes through all vertices.

- ***Shortest path*** **from *u* to *v***: The path from ***u*** to ***v*** with the minimum length (assuming ***u*** is reachable from ***v***).
- ***Distance d<u,v>***: The length of the shortest path from ***u*** to ***v***. If ***u*** is not reachable from ***v***, then by convention, $<u,v>=\infty$.

- **Properties of a Distance Function *d<u,v>* :**
  - $d<u,v> \geq 0$ and $d<u,v>=0 \Leftrightarrow u=v$
  - $d<u,v>+d<v,w> \geq d<u,w>$
  - **Note:** Distance is ***not symmetric***

Objective：

Key Concepts：