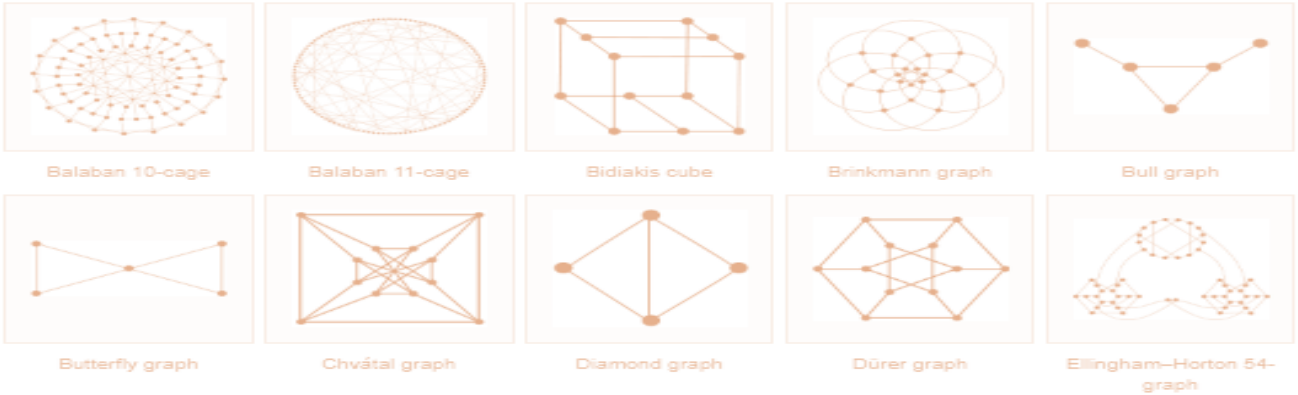
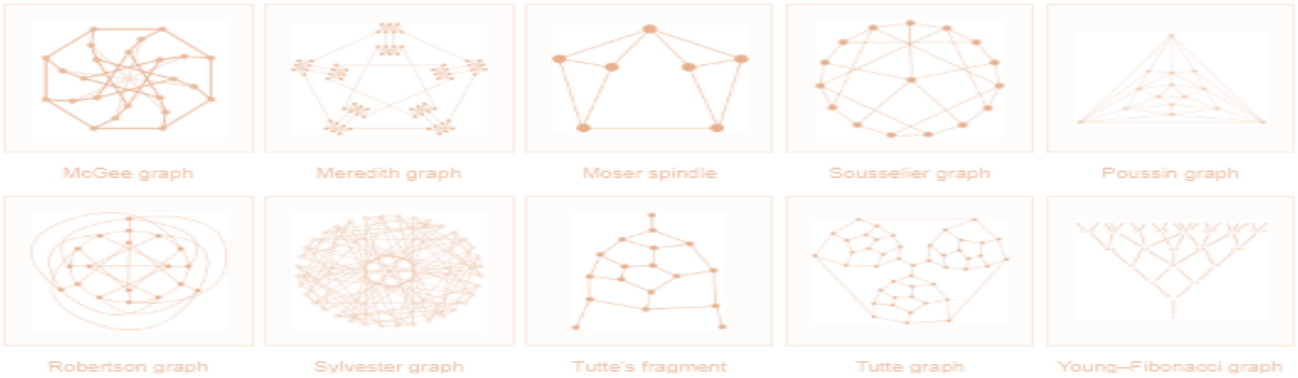
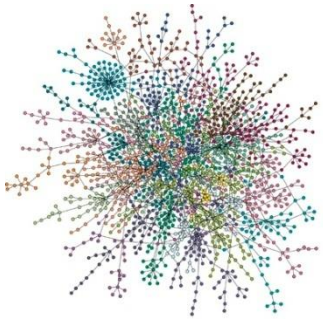


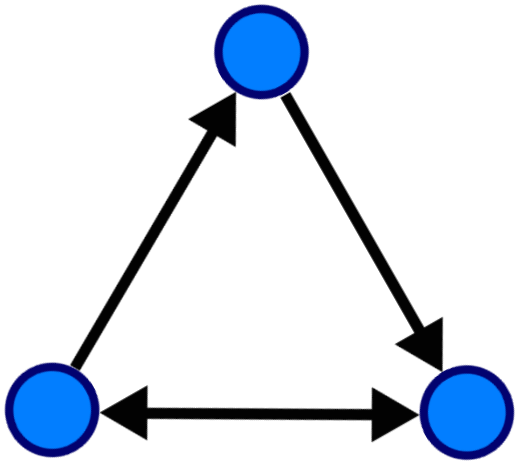
Graph Theory

Directed Graphs

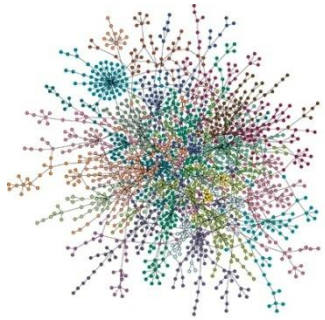


Planarity





Directed Graphs



Directed Graphs Basics

- A **directed graph** (or **digraph**) or **oriented graph** is called a graph $D\ (V, A)$ consisting of a non-empty set of vertices V and a set A of ordered pairs of vertices called arcs.
- In an arc (v, w) the vertices v and w are called **tail** and **head**, or, **source** and **sink**, respectively.
- **In-neighborhood** of a vertex v is the sum of u vertices defined as:

$$N^{-}(v) = \{u \in \mathrm{V}(N) | (u, v) \in \mathrm{A}(D)\}$$

- **Out-neighborhood** of a vertex v is the sum of u vertices defined as:

$$N^{+}(v) = \{u \in \mathrm{V}(N) | (v, u) \in \mathrm{A}(D)\}$$

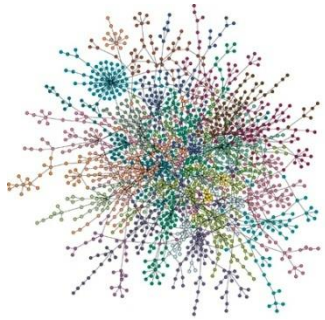
- **In-degree** of a vertex $v, d^{-}(v)$, is the number of arcs having v as head.

$$d^{-}(v) = |N^{-}(v)|$$

- **Out-degree** of a peak $v, d^{+}(v)$, is the number of arcs having v as tail.

$$d^{+}(v) = |N^{+}(v)|$$

Directed Graphs



Directed Graphs Basics

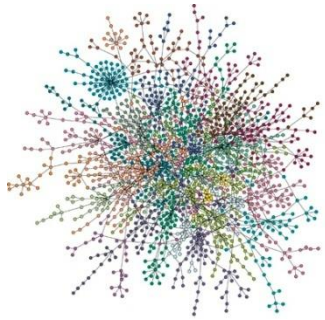
- A **directed graph** (or **digraph**) or **oriented graph** is called a graph $D\ (V, A)$ consisting of a non-empty set of vertices V and a set A of ordered pairs of vertices called arcs.
- The minimum and maximum in-degree of a graph are denoted by $d^-(G)$ and $d^-(\ G)$, respectively, and the minimum and maximum out-degree of a graph are denoted by $d^+(G)$ and $D^+(G)$.
- A digraph is called **balanced**, or **pseudosymmetric**, or **isograph**, if for each vertex v it holds: $d^-(v) = d^+(v)$, or equivalently , $N^-(v) = N^+(v)$.

Handshake Lemma for Directed Graphs

For each directed graph it holds the equation:

$$\sum_{i=1}^{n} d^-(v) = \sum_{i=1}^{n} d^+(v)$$

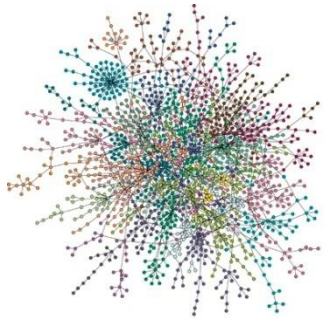
Directed Graphs



Directed Graphs Basics

- A **directed graph** (or **digraph**) or **oriented graph** is called a graph $D\ (V, A)$ consisting of a non-empty set of vertices V and a set A of ordered pairs of vertices called arcs.
- The minimum and maximum in-degree of a graph are denoted by $d^-(G)$ and $d^-(\ G)$, respectively, and the minimum and maximum out-degree of a graph are denoted by $d^+(G)$ and $D^+(G)$.
- A digraph is called **balanced**, or **pseudosymmetric**, or **isograph**, if for each vertex v it holds: $d^-(v) = d^+(v)$, or equivalently , $N^-(v) = N^+(v)$.
- A digraph is called **simple** if no loops and parallel arcs are permitted
- If in a digraph loops are allowed, but parallel/multiple arcs are not allowed then the graph is called **asymmetric** or **antisymmetric**.
- A digraph is called **symmetric** if for each arc (u, w) there exist an arc (w, u).
- A complete asymmetric graph without loops called **tournament**.

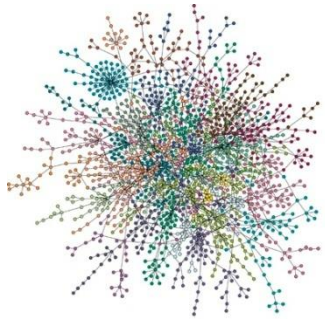
Directed Graphs



Directed Graphs Basics

- A **directed graph** (or **digraph**) or **oriented graph** is called a graph $D\ (V, A)$ consisting of a non-empty set of vertices V and a set A of ordered pairs of vertices called arcs.
- The minimum and maximum in-degree of a graph are denoted by $d^-(G)$ and $d^-(\ G)$, respectively, and the minimum and maximum out-degree of a graph are denoted by $d^+(G)$ and $D^+(G)$.
- A graph is called **underlying** if it results from a digraph by replacing its arcs with undirected edges.
- The **converse** of a digraph is obtained by inverting the direction of its arcs.
- A **complete** digraph is the graph where every pair of vertices is joined by a unique arc.

Directed Graphs

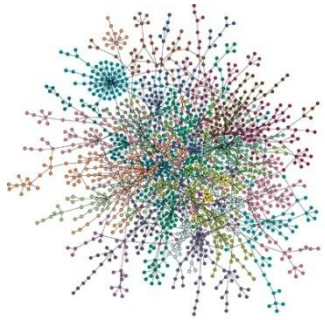


Connectivity in Directed Graphs

- A dirgraph D is **weakly connected**, if the corresponding underlying graph is connected.
- A dirgraph D is **unilateral**, if any two vertices v and w are connected with an oriented path (v, w) OR an oriented path (w, v)
- A dirgraph D is **strongly connected**, if any two vertices v and w are connected with an oriented path (v, w) AND an oriented path (w, v)
- If a dirgraph D is strongly connected or unilateral, then it holds that D is also weakly connected, obviously the opposite does not hold.
- A connected (non-directed) graph G is called **orientable** if its edges can be oriented, so that the corresponding digraph D to be strongly connected.

- **Theorem 1:**
  A connected graph is orientable *iff* each of its edges is contained in at least one cycle.

Directed Graphs



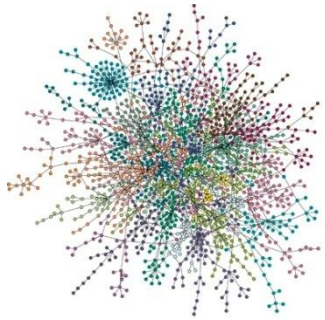
Connectivity in Directed Graphs

- A dirgraph D is **weakly connected**, if the corresponding underlying graph is connected.
- A dirgraph D is **unilateral**, if any two vertices v and w are connected with an oriented path (v, w) OR an oriented path (w, v)
- A dirgraph D is **strongly connected**, if any two vertices v and w are connected with an oriented path (v, w) AND an oriented path (w, v)
- If a dirgraph D is strongly connected or unilateral, then it holds that D is also weakly connected, obviously the opposite does not hold.
- A connected (non-directed) graph G is called **orientable** if its edges can be oriented, so that the corresponding digraph D to be strongly connected.

- **Theorem 1:**
  A connected graph is orientable *iff* each of its edges is contained in at least one cycle.

  Robbins 1939: A graph is orientable *iff* it is connected and has no bridge.

Directed Graphs



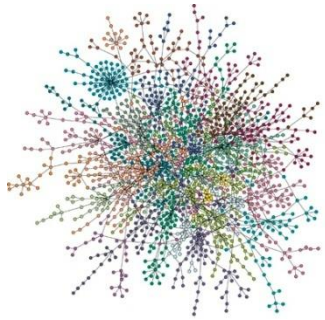
Connectivity in Directed Graphs

- A dirgraph D is **weakly connected**, if the corresponding underlying graph is connected.
- A dirgraph D is **unilateral**, if any two vertices v and w are connected with an oriented path (v, w) OR an oriented path (w, v)
- A dirgraph D is **strongly connected**, if any two vertices v and w are connected with an oriented path (v, w) AND an oriented path (w, v)
- If a dirgraph D is strongly connected or unilateral, then it holds that D is also weakly connected, obviously the opposite does not hold.
- A connected (non-directed) graph G is called **orientable** if its edges can be oriented, so that the corresponding digraph D to be strongly connected.

- **Theorem 2 (Ryser 1957, Fulkerson 1965):**
  A sequence of unordered pairs of non-negative integers $S: (d_1^-, d_1^-), (d_2^-, d_2^+,), \dots, (d_n^-, d_n^+)$, where $d_1^- \geq d_2^- \geq \cdots, \geq d_n^-$, is graphical and corresponds to a directed a directed graph *iff* for each integer k there holds the relations: $d_n^- \leq n-1$, $d_n^+ \leq n-1$ and $\sum_{i=1}^n d_i^- = \sum_{i=1}^n d_i^+$,
  while for $1 \leq j \leq n$ it holds:

$$\sum_{k=1}^n d_k^- \leq \sum_{k=1}^j \min(j-1, d_k^+) + \sum_{k=j+1}^n \min(j, d_k^+).$$

Directed Graphs



Tournament Graphs

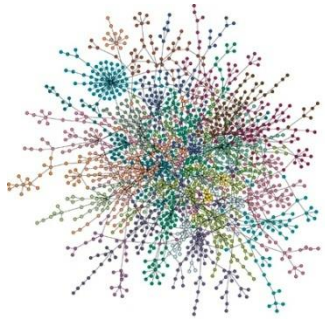
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^-(v) = \sum_{v \in V} d^+(v)$$

- **Theorem 3:**
  Let v be a vertex with maximum out-degree in a tournament graph. The distance from this vertex to any other vertex is 1 or 2.

  - Suppose that vertex v has a maximum out-degree $d^+(v) = k$ and suppose that it is adjacent to vertices $v_1, v_2, \ldots, v_k$.
  - Thus, the distance of v to these vertices is $dist(v, v_i) = 1, \forall\ 1 \leq i \leq k$.
  - In addition, v is adjacent to the remaining $n - k - 1$ vertices, that we denote as $u_1, u_2, \ldots, u\_(n-k-1)$.
  - It must be shown that $dist\ (v, u_i) = 2, \forall\ 1 \leq i \leq\ n - k - 1$.

Directed Graphs



Tournament Graphs

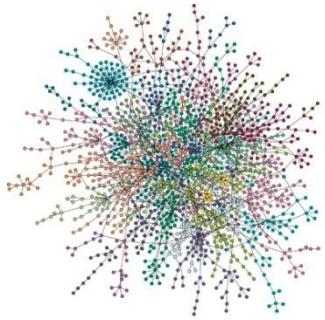
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- **Theorem 3:**
  Let v be a vertex with maximum out-degree in a tournament graph. The distance from this vertex to any other vertex is 1 or 2.

  - If every vertex u_i $(1 \leq i \leq n - k - 1)$ is adjacent $(dist\,(v_j, u_i) = 1)$ to some vertex v_j $(1 \leq j \leq k)$, then the proposition holds.
  - Suppose some top u_l $(1 \leq l \leq n-k-1)$ is not adjacent to any vertex v_i.
  - Then u is adjacent to all peaks $v_1, v_2, \ldots, v_k$ and is also adjacent to v.
  - But then $d^{+}(u_l) = k + 1$ would be inappropriate because then u would have a greater degree outside of v.
  - Thus, every vertex u_l is adjacent to some vertex v_i.

Directed Graphs



Tournament Graphs

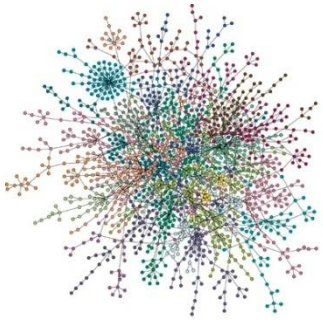
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 4:**
  A tournament graph is transitive *iff* it is contains no cycles.
  ($\Rightarrow$)
  - Let T a tournament graph of n vertices.
  - Let us assume that T contains cycles $C = (x_1, x_2, \ldots, x_r, x_1)$: $r \geq 3$.

Directed Graphs



Tournament Graphs

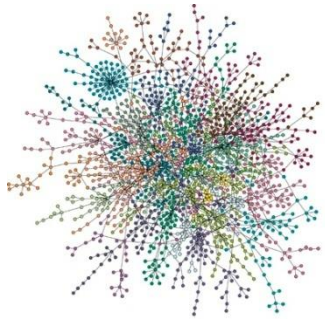
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 4:**
  A tournament graph is transitive *iff* it is contains no cycles.
  ($\Rightarrow$)
  - Since there exist the arcs (x_1, x_2) and (x_2, x_3) due to transitive orientation property there should also exist the arc (x_1, x_3).

Directed Graphs



Tournament Graphs

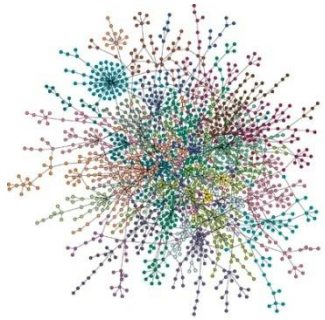
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 4:**
  A tournament graph is transitive *iff* it is contains no cycles.
  (⇒)
  - Similarly, it is implied that there should also exist the arcs $(x_1, x_4), \ldots (x_1, x_r)$.

Directed Graphs



Tournament Graphs

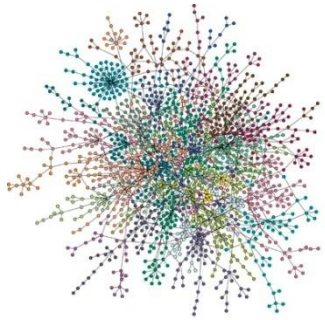
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 4:**
  A tournament graph is transitive *iff* it is contains no cycles.
  ($\Rightarrow$)
  - However, it is a contradiction since there also exist the arc (x_r, x_1).
  - Thus, T should contain no cycle.

Directed Graphs



Tournament Graphs

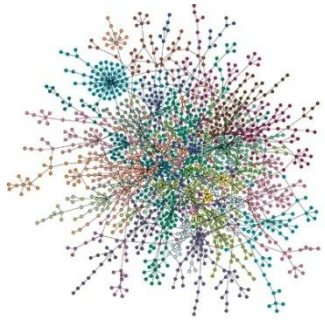
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 4:**
  A tournament graph is transitive *iff* it is contains no cycles.
  ($\Leftarrow$)
  - Let us assume that T is an acyclic tournament graph and that there exist the arcs (x_1, x_2) and (x_2, x_3).
  - Since T contains no cycle it is implied that there does not exist arc (x_3, x_1)
  - Thus, there exist arc (x_1, x_3) and the graph is transitive.

Directed Graphs



Tournament Graphs

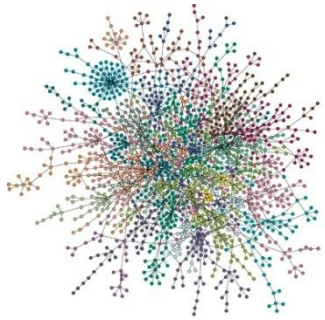
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 5:**
  A non-decreasing sequence S of n non negative integers is graphical score sequence of a transitive tournament graph iff S is : $0,1,2,\ldots,n-1$
  ($\Rightarrow$)
  - Let a tournament graph $T(V, A)$, : $V = \{x_1, x_2, \ldots, x_n\}$ and
    $$A = \{(x_i, x_j) : 1 \leq \mathrm{j} < i \leq \mathrm{n}\}.$$
  - Then, for each vertex it holds that $d^{+}(x_i) = i - 1$ $(1 \leq \mathrm{i} \leq \mathrm{n})$, and hence T is transitive and has S as score sequence.

Directed Graphs



Tournament Graphs

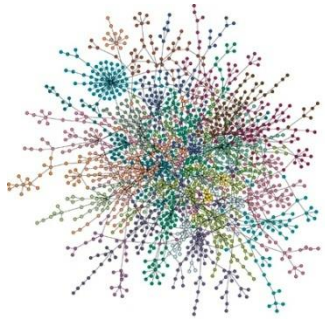
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$m = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^-(v) = \sum_{v \in V} d^+(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 5:**
  A non-decreasing sequence S of n non negative integers is graphical score sequence of a transitive tournament graph iff S is : $0,1,2, \ldots, n-1$
  $(\Leftarrow)$
  - Let us assume that tournament graph T of n vertices is transitive
  - It is sufficient to prove that there do not exist two vertices with the same score.

Directed Graphs



Tournament Graphs

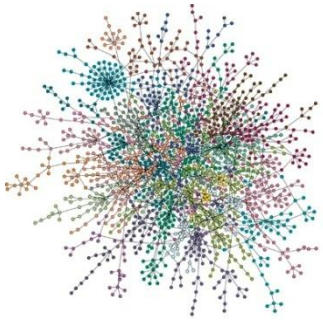
- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 5:**
  A non-decreasing sequence S of n non negative integers is graphical score sequence of a transitive tournament graph iff S is : $0,1,2, \ldots, n-1$
  $(\Leftarrow)$
  - Let us assume that there exist the arc (x_i, x_j) and let the set $N^{+}(x_j)$.
  - Since there exist the arc (x_i, x_j) and the exist arc (x_j, x_k) for each vertex $x_k \in N^{+}(x_j)$, it is implied that there exist arc (x_i, x_k).

Directed Graphs



Tournament Graphs

- Complete directed graph is the graph where each pair of vertices are joined by a unique arc, where a complete asymmetric graph with no loops is a **tournament graph**. The out-degrees are also referred as **score**.

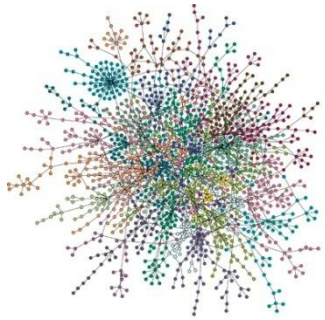
$$\mathrm{m} = \binom{n}{2} = n(n-1)/2 = \sum_{v \in V} d^{-}(v) = \sum_{v \in V} d^{+}(v)$$

- A tournament graph is called a **transitive** if it has the transitive orientation property, i.e., for each pair of arcs (a, b), (b, c) there exist an arc (a, c). [Transitive Triplets and Cyclic Triplets]

- **Theorem 6:**
  A non-decreasing sequence of n non negative integers $S: d_1, d_2, \dots, d_{d_n}, d_{d_{n+1}} - 1, \dots, d_{n-1} - 1$ is a graphical sequence of scores.

- **Theorem 7 (H.G. Landau 1953):**
  A non-decreasing sequence of n non negative integers $S: d_1, d_2, \dots, d_n$ is a graphical sequence of scores *iff* for each integer j $(1 \leq \mathrm{j} \leq \mathrm{n})$ it holds:
  $\sum_{i=1}^{j} d_i \geq \binom{j}{2}$, where the equality holds for $j = n$.

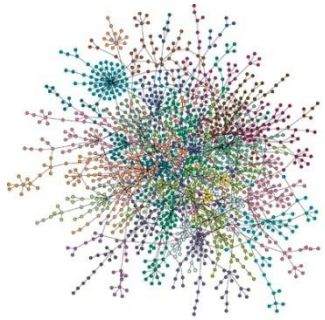
Directed Graphs



Directed Paths and Cycles

- A path in a digraph between two vertices s and t is a sequence of vertices such that for any consecutive pair of vertices u and v to exist the arc (u, v) (forward-arc) or the arc (v, u) (backward arc).
- A directed path is consisted only by forward arc
- A connected digraph D is Eulerian if there exist a directed circuit containing each arc of D.
- In order for a digraph D to be Eulerian it must:
  - be strongly connected
  - not contain sources and sinks, i.e., to not contain vertices with in-(resp. out-) degree equal to 0.
- If the graph is unilateral and does not contain sources or sinks, then it contains an open Eulerian trace.

Directed Graphs



Directed Paths and Cycles

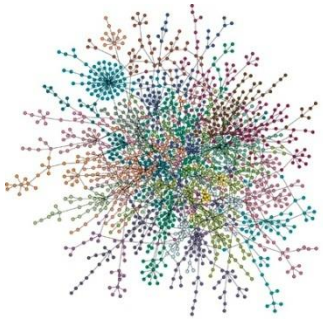
- **Theorem 8 (Ghouila-Houri, 1960):**
  A strongly connected digraph is Eulerian *iff* it is balanced.

- **Corollary:**
  A strongly connected digraph has an Eulerian trace if there exist two vertices v and u such that $d^-(v) = d^+(v) + 1$ and $d^-(u) = d^+(u) + 1$, while for any other vertex w it holds $d^-(w) = d^+(w)$.

- A connected digraph D is Hamiltonian (resp. semi-Hamiltonian), if there exist a directed cycle (resp. path) containing every vertex of D.

- **Theorem 9:**
  A strongly connected digraph D of n vertices is Hamiltonian if $d^-(v) \geq \frac{n}{2}$ and $d^+(v) \geq \frac{n}{2}$ for each vertex $v \in V(D)$

Directed Graphs



Directed Paths and Cycles

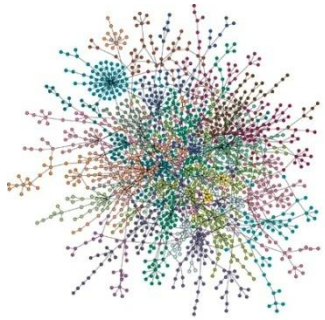
- **Theorem 10 (Redei, 1934):**
  Every tournament graph has a Hamiltonian path, i.e., it is semi-Hamiltonian

- **Theorem 11 (Harray and Moser, 1966):**
  Every strongly connected tournament graph of $n \geq 3$ vertices has cycles of length $l = 3,4, \ldots, n$ (i.e., it is Hamiltonian)

- **Theorem 12 (Roy, 1967 and Gallai, 1968):**
  A digraph D contains directed path of length $x - 1$.

- **Corrolary:**
  Every tournament graph has a Hamiltonian path.

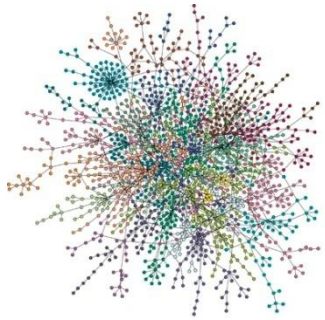
Directed Graphs



DFS on Directed Graphs

- The DFS algorithm is similar to directed graphs except that produced edges are classified into 4 sets:
  - The set T of the edges contained in the trees of the forest and are called **tree-edges**.
  - The set B of the edges called **back-edges** and join descendant vertices to their ancestors.
  - The set F of the edges called **forward-edges** and join ancestral vertices to their descendant .
  - The C set of edges called **cross-edges** and have no ancestor-descendant relationship

Directed Graphs

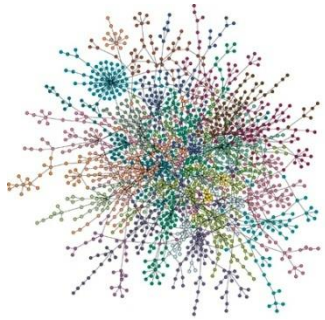


DFS on Directed Graphs

- **Theorem 13:**
  If during the depth first search of a directed graph a cross-edge (u, v) is produced, then it holds: $dfi(u) > dif(v)$

  - Suppose $dfi\,(u) < dfi\,(v)$, visiting vertex u before vertex v.
  - If we assign value in $dfi\,(v)$ when following the edge (u, v), then the edge (u, v) is a tree-edge.
  - Otherwise we visit vertex v as the descendant of vertex u, but not as her son.
  - Thus, the edge (u, v) can not be a cross-edge and therefore we coclude to the contradiction.

Directed Graphs



DFS on Directed Graphs

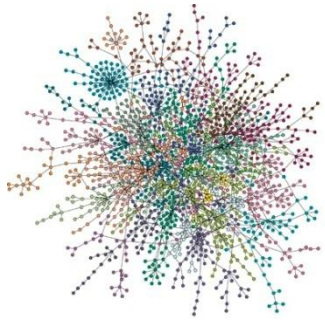
- **Measuring Strong Connectivity deploying DFS:**
  In order to measure the strong connectivity of a digraph we should deploy two times the DFS algorithm starting from an arbitrary vertex..

  - The first time for each vertex v check the unmarked neighboring vertices where v is joined only through outgoing arcs.

  - The second time, for each vertex v we check the unmarked neighboring vertices where v is joined only through incoming arcs.

  - In order for the graph to be strongly connected, all the vertices there should be visited in both applications (first and second time) of the algorithm.

Directed Graphs



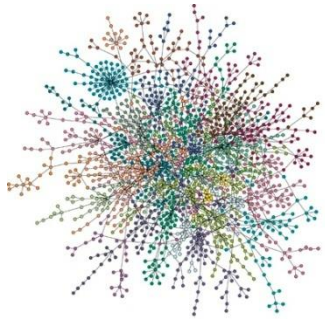
Topological Sorting

- A digraph is acyclic if it does not contain directed circles.
- If during the construction of the DFS tree back-edges are resulting, then we conclude that the graph has a cycle. Conversely, if there is no back-edge the graph is acyclic.

- **Theorem 13:**
  If the edge (u, v) is a back-edge, then it easily results that the graph has a cycle regarding the path from vertex v to vertex u.

  - Let that there exist the cycle $(v_1 \rightarrow v_2 \rightarrow \cdots v_k \rightarrow v_1)$.
  - Let us assume that we first meet the vertex v_i, with the min dfi value.
  - Next, we visit the vertices that have greater dfi values.
  - This way, we will meet the vertex v_{i-1} that leads to vertex v_i, i.e., to a predecessor of v_i, that means that it is a back-edge.

Directed Graphs



Topological Sorting

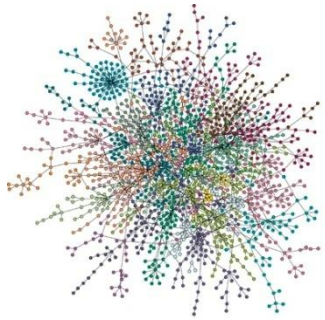
- A digraph is acyclic if it does not contain directed circles.
- If during the construction of the DFS tree back-edges are resulting, then we conclude that the graph has a cycle. Conversely, if there is no back-edge the graph is acyclic.

- **Topological sorting** (or, topological order) of Directed Acyclic Graph is an arrangement the vertices of the grpah such that if the graph G contains the arc (v, w), then vertex v appears before vertex w in the topological order.

- A topological order of G is an arrangement of its vertices $(v_1, v_2, \dots, v_n)$:

$$\forall\, (v_i, v_j) \in G \rightarrow i \; < j \; .$$

Directed Graphs



Topological Sorting

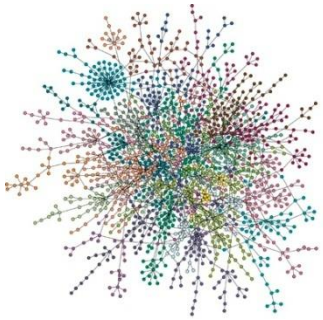
- A digraph is acyclic if it does not contain directed circles.
- If during the construction of the DFS tree back-edges are resulting, then we conclude that the graph has a cycle. Conversely, if there is no back-edge the graph is acyclic.

- **Topological sorting** (or, topological order) of Directed Acyclic Graph is an arrangement the vertices of the grpah such that if the graph G contains the arc (v, w), then vertex v appears before vertex w in the topological order.

- **Algorithm Topological Order with DFS**
  Input: A DAG $D(V, E)$
  Output: Linearization of the vertices of D
  1. Apply DFS on DAG D
  2. Set to each vertex the value $post$ during their extraction from the stack
  3. Sort in descending order the $post$ values.

  .

Directed Graphs



Discovering Strongly Connected Components

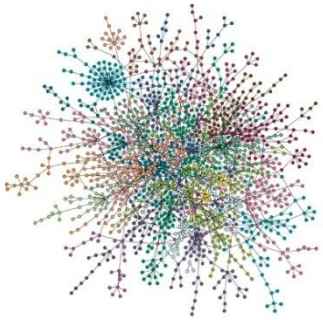
- Kosaraju Algorithm for Discovering Strongly Connected Components.

  Given a directed graph D

  1. Apply DFS enumerating the vertices one-by-one with post values.
  2. Construct the directed graph D' inverting the directions of the arcs of D
  3. Apply DFS on D' starting from the vertex with the max $post$ value. [if the graph has n vertices then start from the vertex with $post$ value $=n$]
  4. Perform a second DFS pass exhausting the graph
  5. Every tree in the resulting DFS forest is a Strongly Connected Component

  .

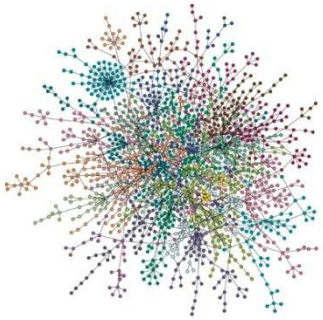
Directed Graphs



Discovering Strongly Connected Components

- Kosaraju Algorithm for Discovering Strongly Connected Components.
- **Algorithm Kosaraju**
  Input: A digraph $D(V, E)$
  Output: The strongly connected components of D
  1. Let S an empty stack.
  2. While S does not contain all the vertices
     1. Perform DFS starting from an arbitrary vertex $v \notin S$
     2. $Push$ (the final vertex u (alongside with all the previous vertices by visit-order)).
  3. D' ← the resulting graph from the inversion of the arcs of D
  4. While S != $isEmpty()$
     1. $Pop(v)$
     2. $DFS(v)$
     3. Pop(visited vertices)
     4. $Delete$ visited vertices from D'

Directed Graphs



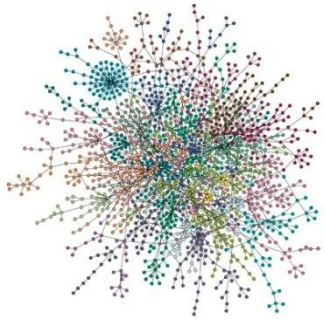
Discovering Strongly Connected Components

- Tarjan Algorithm for Discovering Strongly Connected Components.
- **Algorithm Tarjan SCC**
  Input: A digraph $D(V, E)$
  Output: The vertices of strongly connected components of D
  1. $i \leftarrow 1$.
  2. Truncate the Stack
  3. $\forall\ v \in V$:
     $dfi(v) \leftarrow 0,\ onStack(v) = false$
  4. While $dfi(v) == 0$ for an arbitrary v : StrongComponent(v)

Procedure StrongComponent(v)

1. $dif(v) \leftarrow i,\ L(v) \leftarrow dfi(v),\ i \leftarrow i + 1$
2. $Push(v), onStack(v) \leftarrow true$
3. $\forall\ u \in N^{-}(v)$
   1. If $(dfi(u) == 0)$
      $StrongComponent(u), L(v) \leftarrow \min(L(v), L(u))$
   2. Else If $(dfi(u) < dfi(v)\ AND\ onStack(v) == true$
      $L(v) \leftarrow \min(L(v), dif(u))$
4. If $(L(v) == dfi(v))$
   Pop(vertices "w" until v), $onStack(w) \leftarrow false$.

Directed Graphs

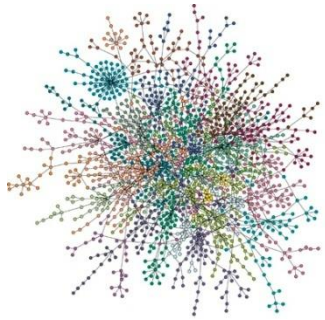


Discovering Strongly Connected Components

- Let $D_1, D_2, \ldots D_k$ the strongly connected components of a digraph D. We define **Condensation** of a digraph D, a digraph D^* with vertices that correspond to the strongly connected components of D, while a pair of vertices of D^* (i.e., two strongly connected components of D) are joined with an arc (D_i, D_j), $i \neq j$, *iff* a vertex in component D_i is adjacent to a vertex in component D_j.

- **Theorem 14:**
  The condensation D^* of any digraph D is an acyclic graph.
  - Let us assume that D^* contains a cycle $D_1, D_2, \ldots D_m, D_1$ $(m \geq 2)$.
  - Let $u \in V(D_1)$ and $v \in V(D_m)$
  - Since D^*contains the path (D_1, D_m) an d the path (D_m, D_1), it is implied that in the directed graph it is included both path (u, v) and (v, u).
  - Thus, vertices u and v belong to the same component D, that is a contradiction, which means that D^* is acyclic.

Directed Graphs



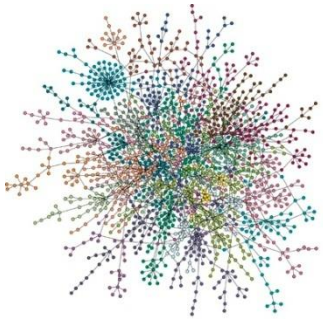
Discovering Strongly Connected Components

- Let $D_1, D_2, \ldots D_k$ the strongly connected components of a digraph D. We define **Condensation** of a digraph D, a digraph D^* with vertices that correspond to the strongly connected components of D, while a pair of vertices of D^* (i.e., two strongly connected components of D) are joined with an arc (D_i, D_j), $i \neq j$, *iff* a vertex in component D_i is adjacent to a vertex in component D_j.

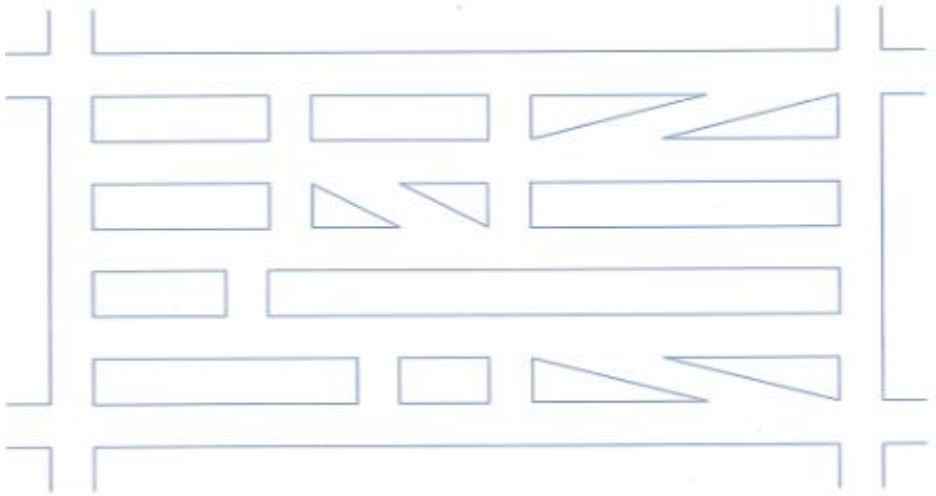
- **Theorem 15:**
  Every directed acyclic graph D contains at least one vertex with in-degree=0 and at least one vertex with out-degree=0

  - Let that $P = (u, \ldots, v)$ a path of D of maximum length.
  - If vertex u is adjacent to a vertex $\in P$, then there results a cycle that contradicts to the main assumption
  - If vertex u is adjacent to a vertex $\notin P$, then there results a path of greater length.
  - Hence the vertex can not be adjacent to any other vertex and it holds $d^-(v) = 0$ (similarly , is proven that $d^+(u) = 0$)

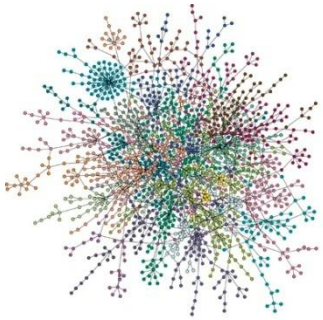
Directed Graphs



- **Problems and Applications**
  1. Can we transform a road network in a one-way street such as the traffic to remain as simple as possible ???



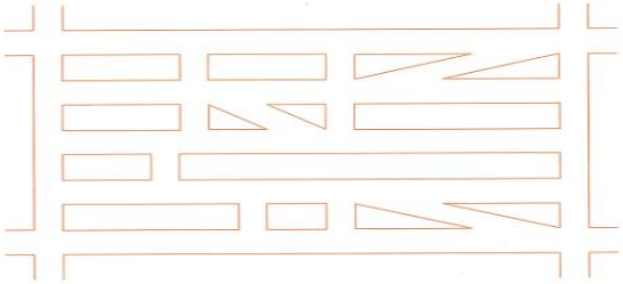
Directed Graphs



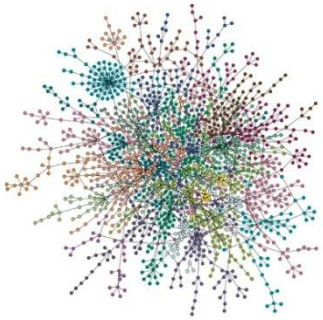
Problems and Applications

1. Can we transform a road network in a one-way street such as the traffic to remain as simple as possible ???

- Transform the road network to a graph:
  - Vertices → Crossroads
  - Edges → Any pair of vertices are joined if we can move from a crossroad to another without crossing any crossroads in the middle.



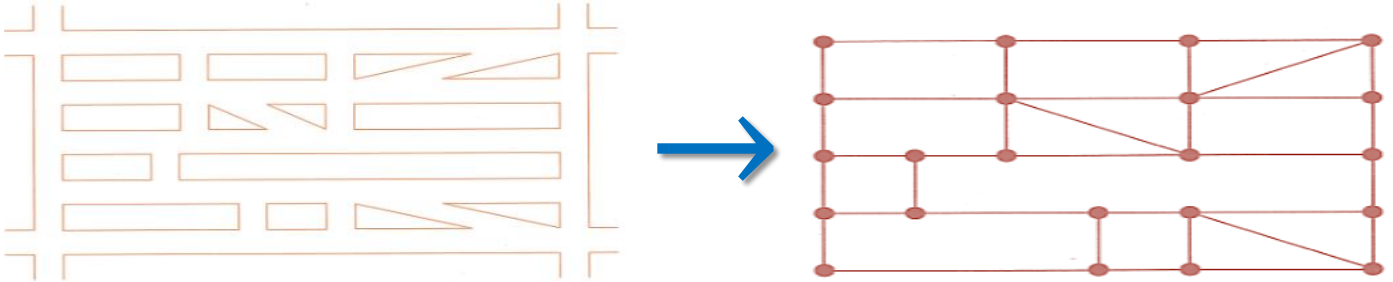
Directed Graphs



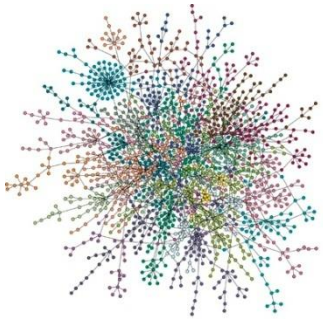
Problems and Applications

1. Can we transform a road network in a one-way street such as the traffic to remain as simple as possible ???

- Transform the road network to a graph:
  - Vertices → Crossroads
  - Edges → Any pair of vertices are joined if we can move from a crossroad to another without crossing any crossroads in the middle.



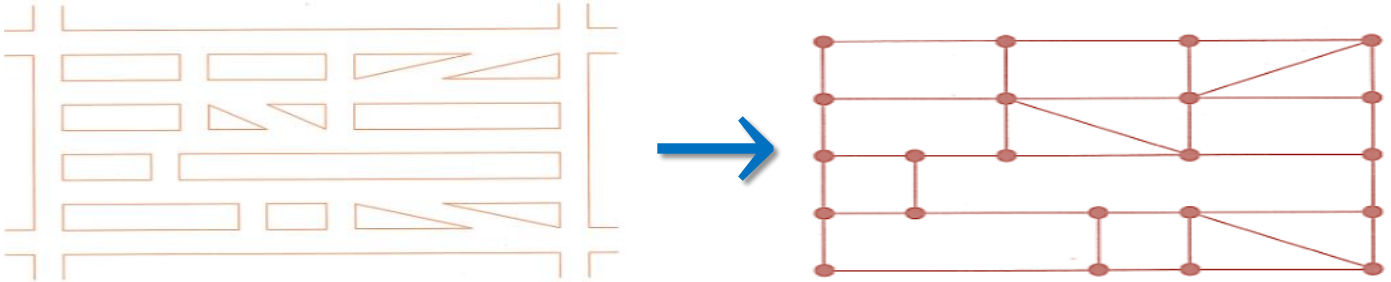
Directed Graphs



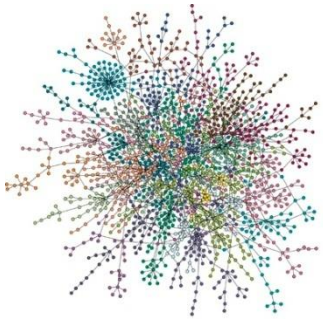
Problems and Applications

1. Can we transform a road network in a one-way street such as the traffic to remain as simple as possible ???

- Transform the road network to a graph:
  - Vertices → Crossroads
  - Edges → Any pair of vertices are joined if we can move from a crossroad to another without crossing any crossroads in the middle.
- Is the graph "orientable" ???



Directed Graphs

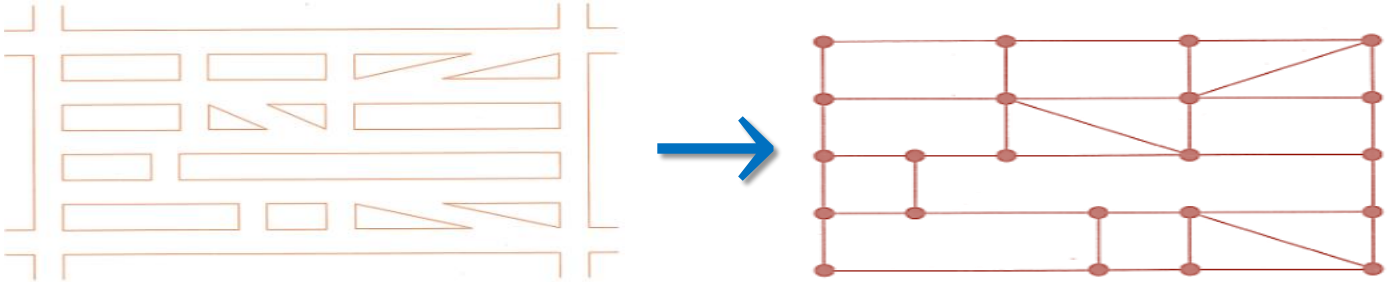


Problems and Applications

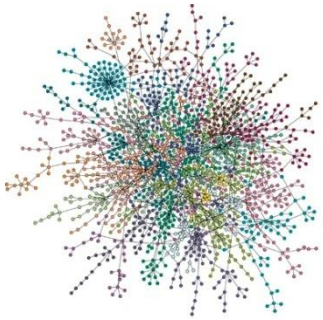
1. Can we transform a road network in a one-way street such as the traffic to remain as simple as possible ???

Robbins, 1939

- Transform the road network to a graph:
  - Vertices → Crossroads
  - Edges → Any pair of vertices are joined if we can move from a crossroad to another without crossing any crossroads in the middle.
  - Is the graph "orientable" ???



Directed Graphs

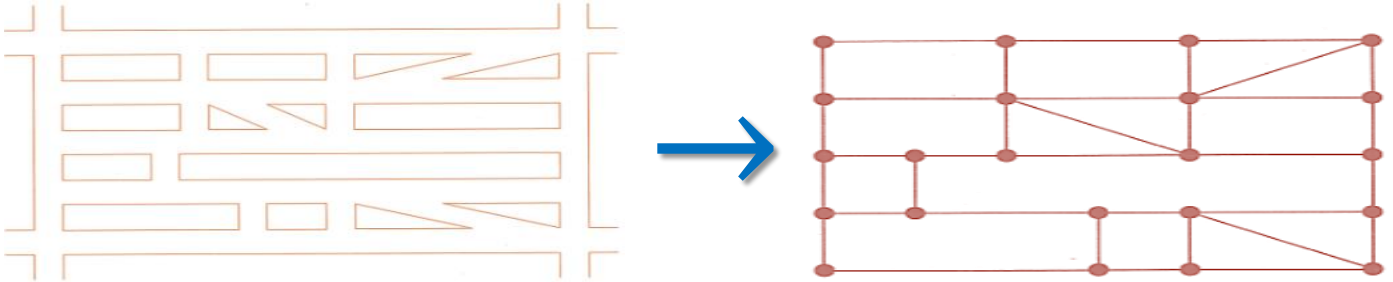


Problems and Applications

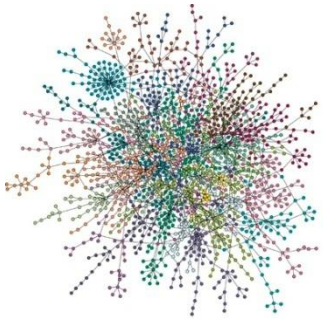
1. Can we transform a road network in a one-way street such as the traffic to remain as simple as possible ???

Robbins, 1939

- Transform the road network to a graph:
  - Vertices → Crossroads
  - Edges → Any pair of vertices are joined if we can move from a crossroad to another without crossing any crossroads in the middle.
  - Is the graph "orientable" ??? [if it contains cut-vertices, the answer is "NO"]



Directed Graphs

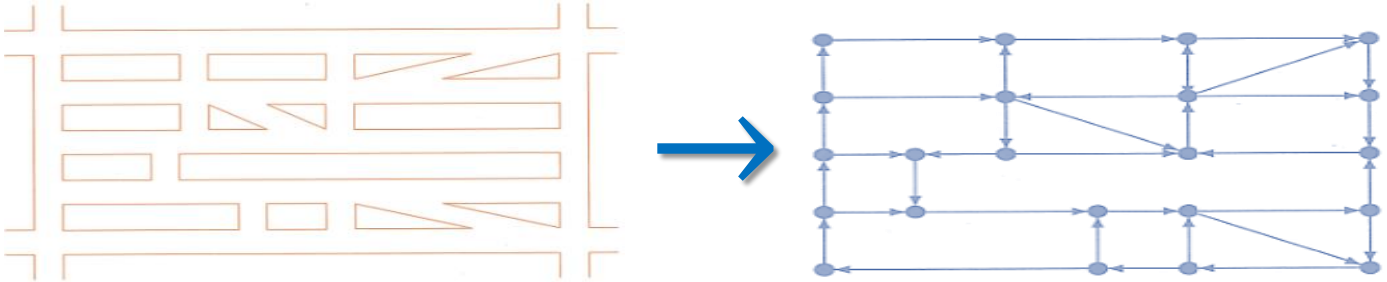


Problems and Applications

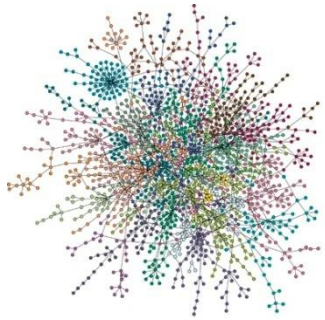
1. Can we transform a road network in a one-way street such as the traffic to remain as simple as possible ???

Robbins, 1939

- Transform the road network to a graph:
  - Vertices → Crossroads
  - Edges → Any pair of vertices are joined if we can move from a crossroad to another without crossing any crossroads in the middle.
  - Is the graph "orientable" ??? [if it contains cut-vertices, the answer is "NO"]



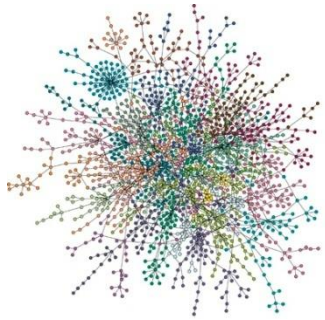
Directed Graphs



- **Problems and Applications**
  2. Can we rearrange the sequence of a set of procedures in order to minimize the time required from a system to process them ???
  - The procedures are denoted by: $J_1, J_2, \ldots, J_n$
  - t_{ij}: preparation time from require for transition from J_i to J_j

	J_1	J_2	J_3	J_4	J_5	J_6
J_1	0	5	3	4	2	1
J_2	1	0	1	2	3	2
J_3	2	5	0	1	2	3
J_4	1	4	4	0	1	2
J_5	1	3	4	5	0	5
J_6	4	4	2	3	1	0

Directed Graphs

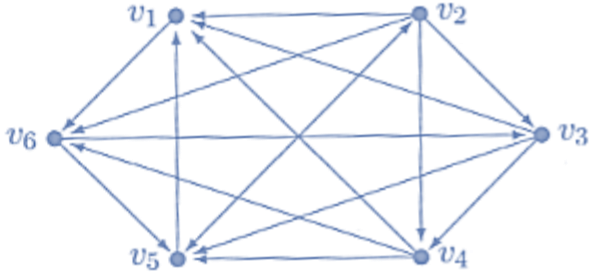


Problems and Applications

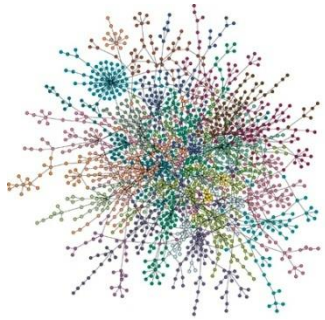
2. Can we rearrange the sequence of a set of procedures in order to minimize the time required from a system to process them ???

- The procedures are denoted by: $J_1, J_2, \ldots, J_n$
- t_{ij}: preparation time from require for transition from J_i to J_j
- Transform the sequence of procedures to a tournament graph
  - Vertices $\rightarrow$ procedures
  - Arcs (i, j) $\rightarrow$ if $t_{ij} \leq t_{ji}$

	J_1	J_2	J_3	J_4	J_5	J_6
J_1	0	5	3	4	2	1
J_2	1	0	1	2	3	2
J_3	2	5	0	1	2	3
J_4	1	4	4	0	1	2
J_5	1	3	4	5	0	5
J_6	4	4	2	3	1	0



Directed Graphs

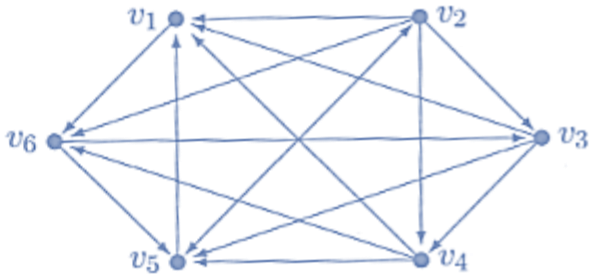


- **Problems and Applications**

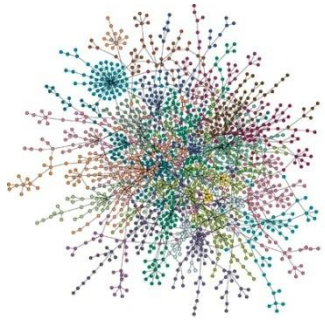
2. Can we rearrange the sequence of a set of procedures in order to minimize the time required from a system to process them ???

- The procedures are denoted by: $J_1, J_2, \ldots, J_n$
- t_{ij}: preparation time from require for transition from J_i to J_j
- Transform the sequence of procedures to a tournament graph
  - Vertices $\rightarrow$ procedures
  - Arcs (i, j) $\rightarrow$ if $t_{ij} \leq t_{ji}$
- Find Hamiltonian cycle with less weight (i.e., TSP)

	J_1	J_2	J_3	J_4	J_5	J_6
J_1	0	5	3	4	2	1
J_2	1	0	1	2	3	2
J_3	2	5	0	1	2	3
J_4	1	4	4	0	1	2
J_5	1	3	4	5	0	5
J_6	4	4	2	3	1	0



Directed Graphs



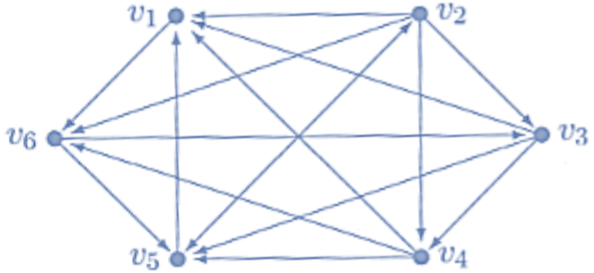
Problems and Applications

2. Can we rearrange the sequence of a set of procedures in order to minimize the time required from a system to process them ???

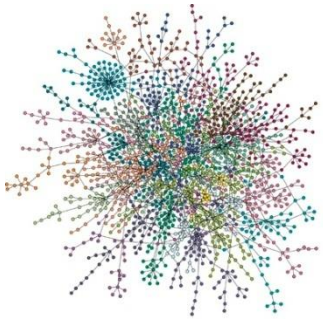
- The procedures are denoted by: $J_1, J_2, \dots, J_n$
- t_{ij}: preparation time from require for transition from J_i to J_j
- Transform the sequence of procedures to a tournament graph
  - Vertices → procedures
  - Arcs (i, j) → if $t_{ij} \leq t_{ji}$
- Find Hamiltonian cycle with less weight (i.e., TSP)

… the graph has a Hamiltonian path!!!
$J_1 \to J_6 \to J_3 \to J_4 \to J_5 \to J_2$

	J_1	J_2	J_3	J_4	J_5	J_6
J_1	0	5	3	4	2	1
J_2	1	0	1	2	3	2
J_3	2	5	0	1	2	3
J_4	1	4	4	0	1	2
J_5	1	3	4	5	0	5
J_6	4	4	2	3	1	0

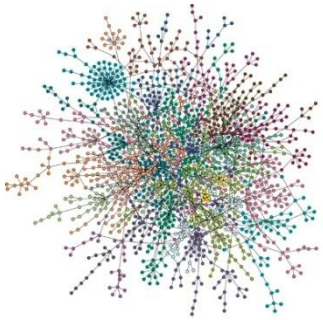


Directed Graphs



- **Problems and Applications**
  3. What is the length of the circular sequence such that no subsequences of r concecutive digits to appear more than once ???

Directed Graphs

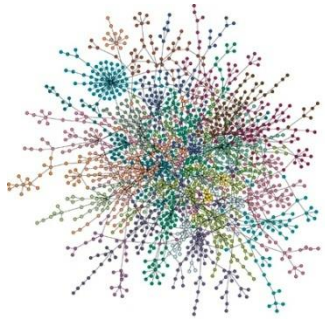


- **Problems and Applications**

3. What is the length of the circular sequence such that no subsequences of r concecutive digits to appear more than once ???

- With r digits there can be produced 2^r discrete subsequences, and hence the circular sequence has length $n \geq 2^r$. We can construct a graph with 2^{r-1} vertices with inscriptions all the possible subsequences of length $r-1$ digits

Directed Graphs

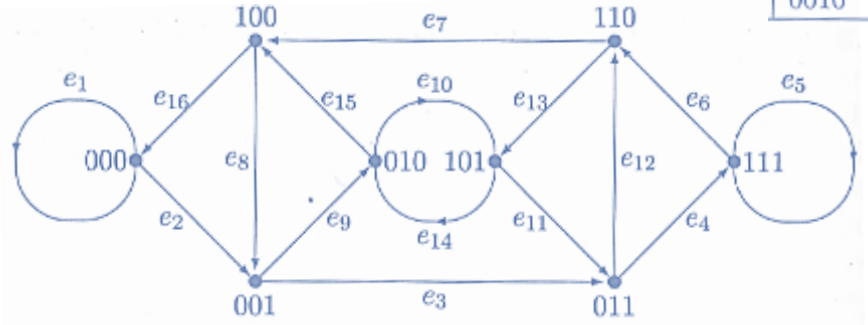


Problems and Applications

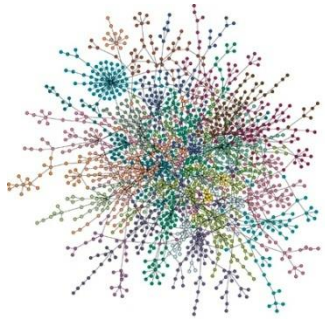
3. What is the length of the circular sequence such that no subsequences of r concecutive digits to appear more than once ???

- With r digits there can be produced 2^r discrete subsequences, and hence the circular sequence has length $n \geq 2^r$. We can construct a graph with 2^{r-1} vertices with inscriptions all the possible subsequences of length $r-1$ digits
- Let that the inscription of a vertex is $a_1 a_2 \ldots a_{r-1} : a_i = 0/1$. Two arcs are directed from this vertex to two vertices with inscriptions $a_2 a_3 \ldots a_{r-1} 0$ and $a_2 a_3 \ldots a_{r-1} 1$, respectively. In these two arcs there been placed the inscriptions $a_1 a_2 a_3 \ldots a_{r-1} 0$ and $a_1 a_2 a_3 \ldots a_{r-1} 0$, respectively.

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
0000	0001	0011	0111	1111	1110	1100	1001
e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}
0010	0101	1011	0110	1101	1010	0100	1000



Directed Graphs

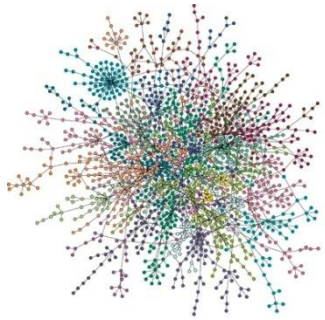


Problems and Applications

3. What is the length of the circular sequence such that no subsequences of r concecutive digits to appear more than once ???

- With r digits there can be produced 2^r discrete subsequences, and hence the circular sequence has length $n \geq 2^r$. We can construct a graph with 2^{r-1} vertices with inscriptions all the possible subsequences of length $r-1$ digits
- Let that the inscription of a vertex is $a_1 a_2 \dots a_{r-1} : a_i = 0/1$. Two arcs are directed from this vertex to two vertices with inscriptions $a_2 a_3 \dots a_{r-1} 0$ and $a_2 a_3 \dots a_{r-1} 1$, respectively. In these two arcs there been placed the inscriptions $a_1 a_2 a_3 \dots a_{r-1} 0$ and $a_1 a_2 a_3 \dots a_{r-1} 0$, respectively.
- If $a_1 = a_2 = a_3 = \cdots = a_{r-1}$ then it is a loop.
- Every vertex has in-degree=out-degree=2, and hence the graph is balanced.
- From Theorem .8 the graph is Eulerian, and further contains path of length $n = 2^r$ arcs.

Directed Graphs

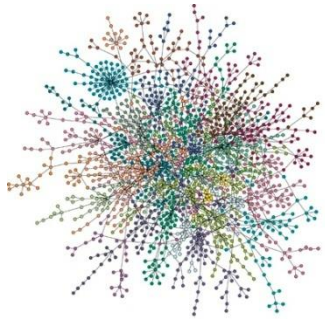


Problems and Applications

3. What is the length of the circular sequence such that no subsequences of r concecutive digits to appear more than once ???

- With r digits there can be produced 2^r discrete subsequences, and hence the circular sequence has length $n \geq 2^r$. We can construct a graph with 2^{r-1} vertices with inscriptions all the possible subsequences of length $r-1$ digits
- Let that the inscription of a vertex is $a_1 a_2 \dots a_{r-1} : a_i = 0/1$. Two arcs are directed from this vertex to two vertices with inscriptions $a_2 a_3 \dots a_{r-1} 0$ and $a_2 a_3 \dots a_{r-1} 1$, respectively. In these two arcs there been placed the inscriptions $a_1 a_2 a_3 \dots a_{r-1} 0$ and $a_1 a_2 a_3 \dots a_{r-1} 0$, respectively.
- If $a_1 = a_2 = a_3 = \cdots = a_{r-1}$ then it is a loop.
- Every vertex has in-degree=out-degree=2, and hence the graph is balanced.
- The last $r-1$ digits of an arc equal the $r-1$ first digits of the next arc.

Directed Graphs



Problems and Applications

3. What is the length of the circular sequence such that no subsequences of r concecutive digits to appear more than once ???

- With r digits there can be produced 2^r discrete subsequences, and hence the circular sequence has length $n \geq 2^r$. We can construct a graph with 2^{r-1} vertices with inscriptions all the possible subsequences of length $r-1$ digits
- Let that the inscription of a vertex is $a_1 a_2 \ldots a_{r-1} : a_i = 0/1$. Two arcs are directed from this vertex to two vertices with inscriptions $a_2 a_3 \ldots a_{r-1} 0$ and $a_2 a_3 \ldots a_{r-1} 1$, respectively. In these two arcs there been placed the inscriptions $a_1 a_2 a_3 \ldots a_{r-1} 0$ and $a_1 a_2 a_3 \ldots a_{r-1} 0$, respectively.
- If $a_1 = a_2 = a_3 = \cdots = a_{r-1}$ then it is a loop.
- Every vertex has in-degree=out-degree=2, and hence the graph is balanced.
- Obtaining the first digit of each arc of this path, the required sequence is constructed, as there do not exist arcs with the same inscription.