EIDMA

Lecture 14

- Hamiltonian graphs cont-d.
- Trees

Definition.

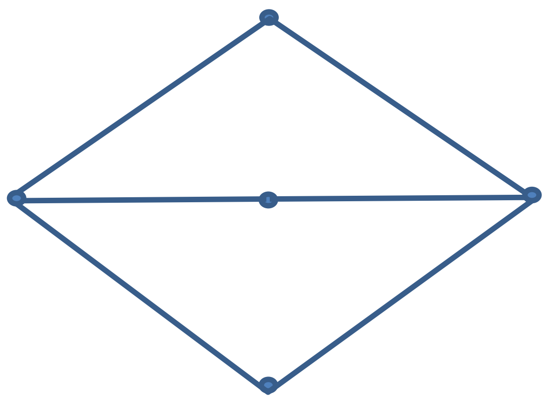
A graph G=(V,E) is called *2-connected* iff |V|>2 and for every vertex *v*, G-{*v*} is connected.

G-{*v*} is often shortened to G-*v*.

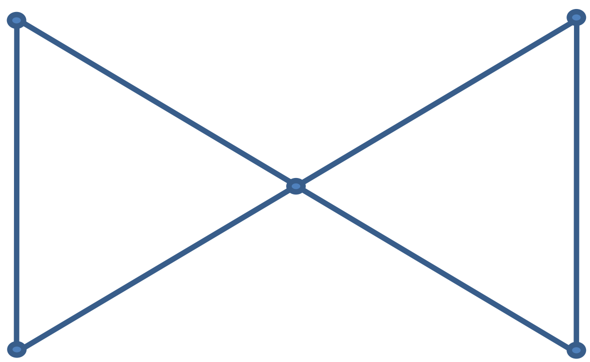
Corollary.

Every Hamiltonian graph is 2-connected.

Example.



2-connected, not Hamiltonian.



Connected, not 2-connected, not Hamiltonian.

SUFFICIENT CONDITIONS

Theorem. (Dirac, 1952)

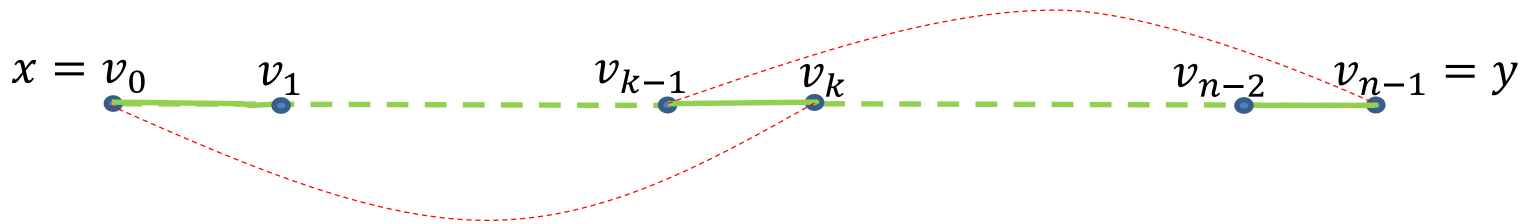
If G has n vertices, $n \geq 3$, and for every vertex v of G $\deg(v) \geq \frac{n}{2}$ then G is Hamiltonian.

Proof.(by contradiction)

Suppose for some $n \geq 3$ there exists a counterexample on n vertices. If there is one, there may be more, we choose one with the largest possible number of edges and denote it by G. This means that for every two nonadjacent vertices x,y, $G+xy$ (xy denotes here the x-y edge) is no longer a counterexample, i.e., either does not satisfy the Dirac's condition or it has a Hamiltonian cycle. Since adding a new edge does not decrease degrees of vertices nor does it decrease the number of vertices the latter must be the case.

Hence, in our hypothetical counterexample every two nonadjacent vertices *x*, *y* are joined by a spanning simple path (v_0 ,v_1, … , v_{n-1}) with $v_0 = x$ and $v_{n-1} = y$.

For every k, if v_k is adjacent to *x* then v_{k-1} is NOT adjacent to *y*.



Otherwise (v_0 , v_k, v_{k+1}, … , v_{n-2}, v_{n-1}, v_{k-1}, v_{k-2},…, v_1, v_0) would be a Hamiltonian cycle in G. So:

- *y* is not adjacent to itself and,
- *y* is not adjacent to the predecessor of a neighbor of *x*.

This means that deg(*y*) ≤ *n* – 1 – deg(*x*), i.e., deg(*y*) + deg(*x*) ≤ *n*–1 which contradicts deg(*y*), deg(*x*) ≥ $\frac{n}{2}$. QED

Theorem. (Ore 1960)

If G has n vertices, $n \geq 3$, and for every two nonadjacent vertices u and v of G, $\deg(u) + \deg(v) \geq n$ then G is Hamiltonian.

Proof.

Essentially the same. The only difference is that you replace

"*Since adding a new edge does not decrease degrees of vertices nor does it decrease the number of vertices …*"

with

"*Since adding a new edge does not decrease sums of degrees of pairs of nonadjacent vertices nor does it decrease the number of vertices …*"

Both Dirac and Ore theorems are pretty heavy-handed. They force a Hamiltonian cycle on G by making sure that G has "many" edges – even though "many" means different things for Dirac and for Ore.

A graph may have as few as n edges and yet be Hamiltonian.

Dirac's condition forces the graph to have at least $\frac{1}{2}n\frac{n}{2} = \frac{n^2}{4}$ edges.

Warning.
As usual: Remember the Dirac and Ore theorems work one way only. They are sufficient but not necessary conditions for the existence of a Hamiltonian cycle in a graph.

TREES

Now let's investigate some cycle-less graph (also called *acyclic* graphs.

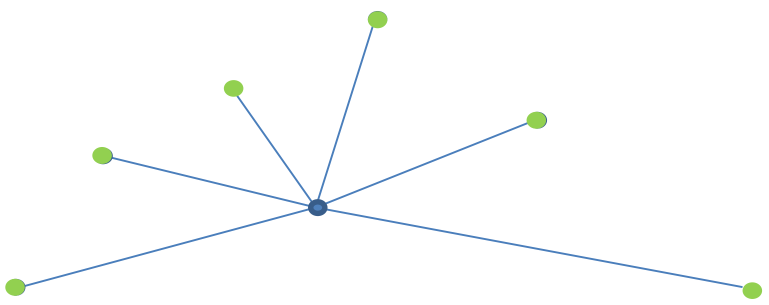
Definition.

A *tree* is a connected graph without cycles. The single-vertex tree is called a *trivial tree.*

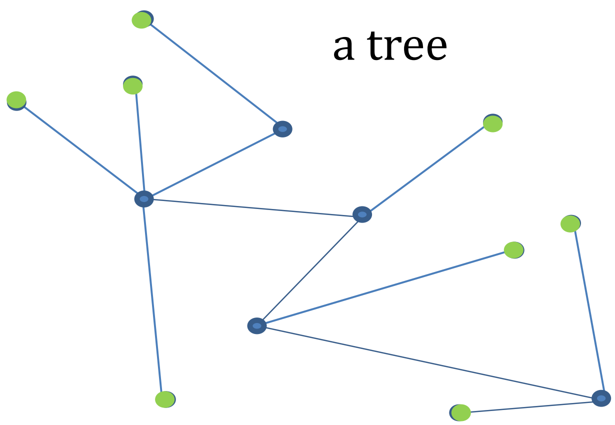
Examples

Paths are trees, K_1 and K_2 are trees. Connected graphs with a single vertex of degree greater than 1 are trees (called *bushes* or *stars*).

Vertices of degree 1 are called *leaves*.



a bush



a tree

Theorem. (Characterisation of trees)

$G = (V, E)$ is a graph. The following conditions are equivalent

1. G is a tree.
2. For each $u, v \in V$ there exists exactly one $u - v$ path in G, and the path is simple.
3. G is connected and $|V| = |E| + 1$.
4. G has no cycles and $|V| = |E| + 1$.
5. G has no cycles and for each nonadjacent $u, v \in V, G + uv$ has exactly one cycle.

Proof. (1⇒2⇒3⇒4⇒5⇒1)

(1⇒2). A $u - v$ path exists because G is connected. A non-simple path contains a simple "sub-path" which also is a $u - v$ path. So, we only have to prove uniqueness. Suppose $(v_0, v_1, \dots, v_k)$ and $(x_0, x_1, \dots, x_p)$ are different $u - v$ paths. Let q be the smallest subscript such that $v_{q+1} \neq x_{q+1}$. Let t be smallest such that $t > q$ and $v_t = x_s$ for some $s > t$. Then $(v_t, v_{t+1}, \dots, x_s, x_{s-1}, \dots, x_{t+1}, x_t = v_t)$ is a cycle in G.

2. For each $u, v \in V$ there exists exactly one u-v path in G, and the path is simple.
3. G is connected and $|V| = |E| + 1$.

(2⇒3). (Induction on $n = |V|$)

For $n = 1$ the equality $|V| = |E| + 1$ is trivially true. Suppose for some $n \geq 1$ the implication is true for all graphs on n vertices and consider a graph G on $n+1$ vertices. G is obviously connected, nontrivial and has no cycles (a cycle would mean 2 paths between 2 of its vertices) . There is at least one vertex, say z, in G of degree 1 (from the Lemma about cycles).

$G - z$ is connected and has exactly one path between any 2 vertices (paths in $G - z$ are also paths in G). So, by the induction hypothesis,

$$|V| = |V(G - z)| + 1 = |E(G - z)| + 1 + 1 = |E| + 1.$$

3. G is connected and $|V| = |E| + 1$.

4. G has no cycles and $|V| = |E| + 1$.

(3⇒4) (Induction on $n = |V|$)

If n=1 G obviously has no cycles.

If n>1 and G satisfies 3 then G has a vertex, say z, of degree 1 (otherwise from the handshaking lemma we have $n - 1 = |E| = \frac{1}{2}\sum \deg(v) \geq n$). G–z is connected and has 1 vertex and 1 edge fewer than G, which means it satisfies 3. so, by the induction hypothesis, G–z has no cycles. Since z does not belong to a cycle, G has no cycles as well.

4. G has no cycles and $|V| = |E| + 1$.
5. G has no cycles and for each nonadjacent $u, v \in V, G + uv$ has exactly one cycle.

(4⇒5). Let $G_1, G_2, \ldots, G_k$ be components of G. Since G has no cycles, each $G_i = (V_i, E_i)$ is a tree so, by 2., $|V_i| = |E_i| + 1$. Adding these side-to-side we get $|V| = \sum|V_i| = \sum(|E_i| + 1) = k + \sum|E_i| = k + |E|$. Since $|V| = |E| + 1$ we get $k = 1$ i.e., G is connected, hence a tree. In a tree every two (nonadjacent or not) vertices u and v are joined by exactly one path. Addition of the edge uv to G results in a graph with exactly one cycle.

5. G has no cycles and for each nonadjacent $u, v \in V, G + uv$ has exactly one cycle.
1. G is a tree.

(5⇒1)

Since $G + uv$ has a cycle and G does not, G has a $u - v$ path. This means G has no cycles and G is connected, hence a tree. QED.

Fact.

Every nontrivial tree has at least 2 leaves.

Proof. We proved in (2 ⇒3) that every nontrivial tree has at least one leaf. Suppose a tree G on n vertices has exactly one, say z. Then other vertices have degrees ≥ 2, so

$$n - 1 = |E| = \frac{1}{2}(1 + \sum_{v \neq z} \deg(v)) \geq \frac{1}{2} + (n - 1) = n - \frac{1}{2}$$

which is obviously not true. QED

Definiton. A graph whose every component is a tree is called (surprise, surprise !) a *forest.*

Fact.

Every connected graph G contains a *spanning tree* (a spanning subgraph which is a tree)

Proof.

Consider a maximal acyclic subgraph H of G. Clearly H is a spanning forest. Suppose H is not a tree (i.e., it is disconnected). Denote by V_1 the vertex set of one component of H and by V_2 the complement of V_1 ($V_2 = V \setminus V_1$). Since G is connected, there is an edge e between a vertex from V_1 and a vertex from V_2. $H + e$ is obviously connected, has no cycles and is larger than H, contrary to our choice of H.

Definition.

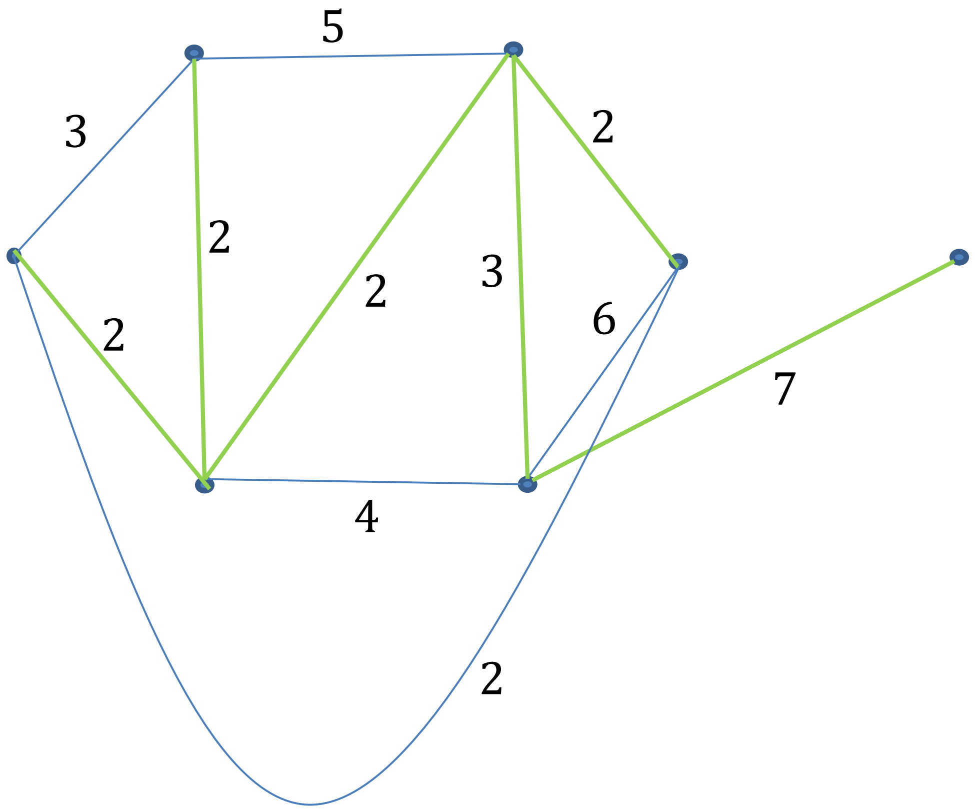
A *weighted* graph N is a pair consisting of a graph $G = (V, E)$ and a function $w: E \to \mathbb{R}^+, (\mathbb{R}^+ = (0; \infty))$. $w(e)$ is referred to as the weight (or the cost) of the edge e.

The *weight of a weighted graph* is the sum of weights of all its edges. This definition extends to subgraphs of a weighted graph as well as the graph itself.

Problem.

Given a connected, weighted graph N, find a spanning tree with the least possible weight (or a cheapest spanning tree if you prefer to think in terms of cost, rather than weight).

Example.



Can we do better?

Solution. Kruskal's algorithm.

$S = \emptyset; i = 1;$

while (there exists an edge e such that the graph consisting of vertices and edges of $S \cup \{e\}$ (i.e., *the subgraph of G induced by* $S \cup \{e\}$), has no cycles)

{

e_i = a cheapest edge such that, the graph induced by $S \cup \{e_i\}$ has no cycles;

$S = S \cup \{e_i\};$

$i = i + 1;$

}

return(S);

Theorem.
The set of edges returned by Kruskal's algorithm induces the cheapest spanning tree in N.
Proof - outline. (by contradiction)
Suppose $T = \{f_1, f_2, \dots, f_s\}$ is the set of edges of a spanning tree of G which is cheaper than S and has as many edges from S as possible. Let p be the smallest subscript such that $e_p \notin T$.Then $T^+ = \{e_p, f_1, f_2, \dots, f_s\}$ has exactly one cycle, say C. At least one edge of C, say f_k does not belong to S – otherwise S would contain the cycle C. The cost of f_k is not smaller than that of e_p – otherwise Kruskal's algorithm in step p would have chosen f_k rather than e_p (the edges $e_1, e_2, \dots, e_{p-1}$ belong both to S and to T, so f_k was admissible at the time. This means that $T + e_p - f_k$ is also cheaper than S but has more edges from S than T does, contrary to our choice of T. QED

Comprehension.

If $N = (G, w\ )$ is a weighted graph and w is a one-to-one function then G has exactly one cheapest spanning tree.