**def** If $T$ is a $k$-ary tree with a vertex $v \in V(T)$, then we define **depth(v)** to be the distance (i.e. length of a shortest walk) in $T$ between $v$ and the root of $T$.

Or, recursively,

$$\text{depth}(v) = \begin{cases} 0 & v \text{ is the root} \\ 1 + \text{depth}(u) & u \text{ is the parent of } v \end{cases}$$

**Compare:** In Lecture 9, we defined the depth of a whole $k$-ary tree. In our new language,

$$\text{depth}(T) = \max_{v \in V(T)} \text{depth}(v)$$

**Thm** The **weight** $w(v)$ of a vertex $v$ in a binary tree $T$ is

$$w(v) = \frac{1}{2^{\text{depth}(v)}}.$$

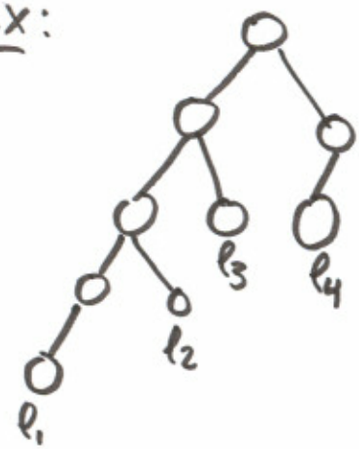
If $T$ is a binary tree and

$$L = \{ v \in V(T) \mid v \text{ has no children} \},$$

then $\sum_{v \in L} w(v) \leq 1$.

(This is called Kraft's inequality.)

**Ex:**




- $L = \{\ell_1, \ell_2, \ell_3, \ell_4\}$

| $v$ | depth($v$) | $w(v)$ |
|---|---|---|
| $\ell_1$ | 4 | $1/16$ |
| $\ell_2$ | 3 | $1/8$ |
| $\ell_3$ | 2 | $1/4$ |
| $\ell_4$ | 2 | $1/4$ |

$11/16 = \sum_{v \in L} w(v)$

Pf By induction on $n = |V(T)|$.

If $n=0$, then $T=()$ is the empty tree, $L=\emptyset$, and

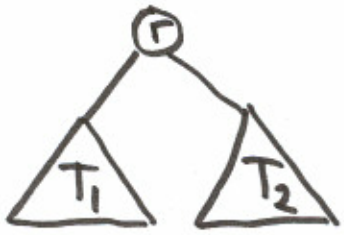
$$\sum_{v \in L} w(v) = 0$$

because there are no terms to sum. Therefore

$$\sum_{v \in L} w(v) = 0 \leq 1$$

and the theorem holds in this case.

Inductive Step: suppose $n \geq 1$; then $T = (r, T_1, T_2)$, where $T_1$ and $T_2$ are binary trees with $|V(T_1)|, |V(T_2)| < n$.




For a vertex $v \in V(T_1)$ (or $v \in V(T_2)$), we use $w_{T_1}(v)$ (or $w_{T_2}(v)$) for the weight of $v$ in the subtree $T_1$ (or $T_2$).

Note: $w(v) = \frac{1}{2} w_{T_1}(v)$, if $v \in V(T_1)$.

Consider two cases: $r \in L$. In this case, $r$ has no children, so $T_1 = T_2 = ()$ (i.e. $T_1$ and $T_2$ are null trees). Therefore $L = \{r\}$ and

$$\sum_{v \in L} w(v) = w(r) = \frac{1}{2^0} = 1 \leq 1$$

so the theorem holds.

In the second case, $r \notin L$. Let

$$L_1 = L \cap V(T_1)$$

$$L_2 = L \cap V(T_2)$$

By the inductive hypothesis,

$$\sum_{v \in L_1} w_{T_1}(v) \leq 1$$

$$\sum_{v \in L_2} w_{T_2}(v) \leq 1 .$$

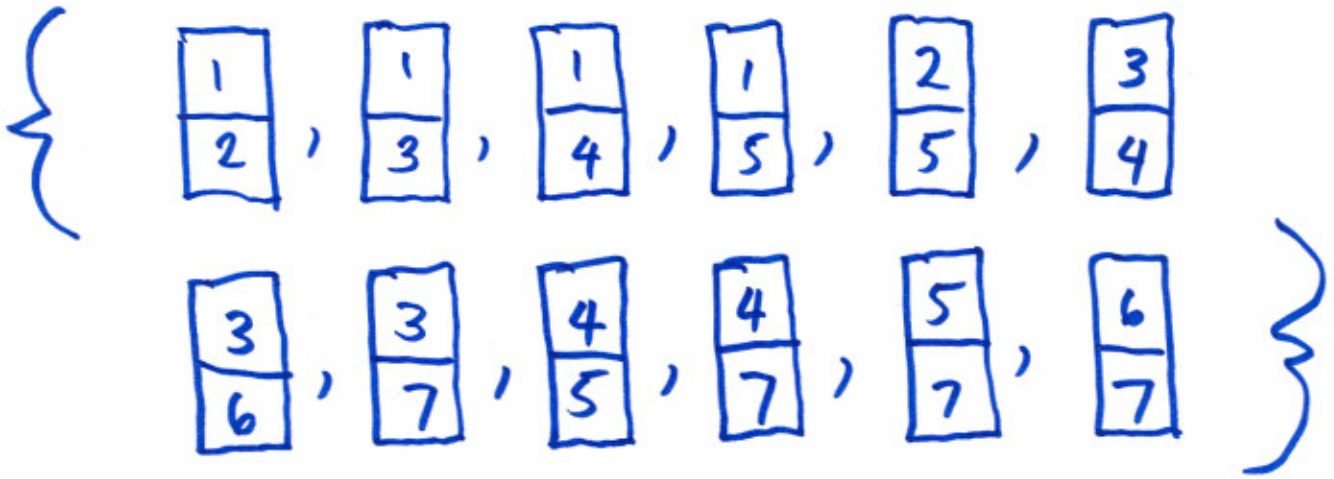
Noting that $L = L_1 \cup L_2$, we have

$$\sum_{v \in L} w(v) = \sum_{v \in L_1} w(v) + \sum_{v \in L_2} w(v)$$

$$= \frac{1}{2} \sum_{v \in L_1} w_{T_1}(v) + \frac{1}{2} \sum_{v \in L_2} w_{T_2}(v) \leq \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1$$

and the theorem holds

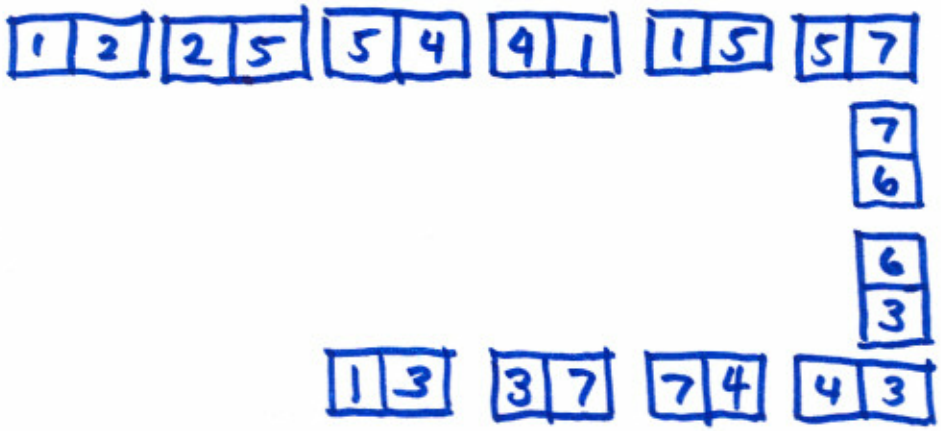
**Problem**: Suppose we are given a set of dominoes:




Assume we are given no duplicates.

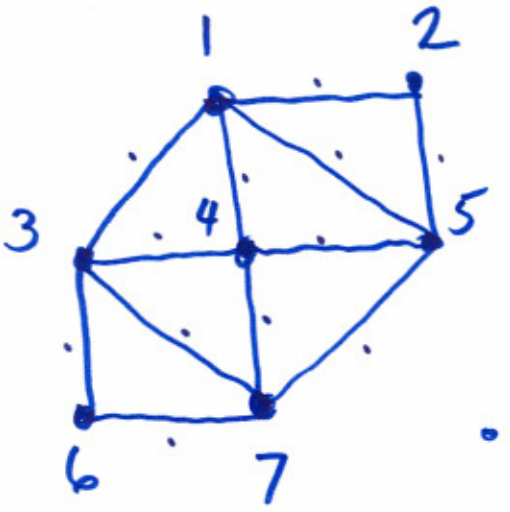
When can we arrange all dominoes in a line, with adjacent dominoes having matching labels?

Eg:




**Answer**: Let us express this as a graph problem. ABSTRACTION

As a graph:




G

$V(G) = \{ k \mid k$ is a number appearing on the dominoes $\}$

$E(G) = \{ \{k_1, k_2\} \mid$ we are given the domino $\begin{smallmatrix} k_1 \\ k_2 \end{smallmatrix} \}$

- In the language of dominoes, we want to know:

  Can we lay the dominoes down in a line with matching endpoints?

- In the language of graphs, we want to know:

  Can we write down the edges of G so that consecutive ~~endpoints~~ edges have matching endpoints?

12, 25, 54, 41, 15, 57, 76, 63, 34, 47, 73, 31

- This is the same as asking:

Is there a walk in G that traverses each edge in G exactly once?

W = 1254157634731

**def** A walk W in a graph G which traverses each edge exactly once is an Eulerian trail. If W is also closed (i.e. starts and ends at the same vertex), then W is an Eulerian circuit.

**Remark:** Recall that a trail is a walk which traverses each edge at most once. So an Eulerian trail is a special kind of trail.

**Ex:** W is an Eulerian circuit in G.
(Here, W = 1254··· and G is on the previous page.)

**Thm** A graph $G$ ~~contains~~ has an Eulerian circuit if and only if

- $G$ has at most one component with edges. (Think: $G$ is connected except, ~~for~~ possibly, for some vertices of degree 0.)
- Each vertex in $G$ has even degree.

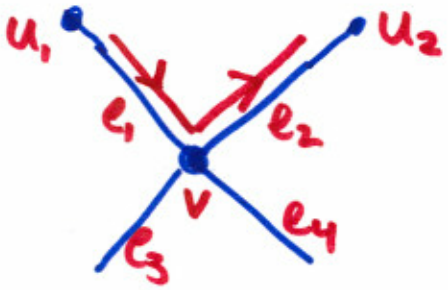
**Pf:** This is an *if and only if* statement, so we have two directions to prove.

($\Rightarrow$) Suppose $G$ has an Eulerian circuit $W$. Because $W$ traverses each edge in $G$, all edges must be in a single component.

Let $v \in V(G)$, and let $S = \{e_1, e_2, \ldots, e_k\}$ be the set of edges incident to $v$.

Each time that $W$ visits $v$, it traverses

two edges in $S$: once while arriving at $v$ and once while leaving $v$.




$W = \cdots u_1 v u_2 \cdots$

Because ~~each edge in~~ these edges are never traversed again and each edge is traversed exactly once,

$2 \cdot$ # visits that $W$ makes to $v$ $= |S|$

Therefore $d(v) = |S|$ is even.

(Question: why don't we need to treat the start/end vertex of $W$ separately?)

($\Longleftarrow$) Suppose $G$ contains at most one component with edges and each vertex in $G$ has even degree.

We show $G$ has an Eulerian circuit.

The proof of this is by induction on $m = |E(G)|$.

Base case: If $m = 0$, $G$ has no edges, so we may pick any vertex $u \in V(G)$ and use $W = u$ as our Eulerian circuit.

Inductive Step: Suppose $m \geq 1$. Because $G$ only has one component $H$ with edges, an Eulerian circuit in $H$ traverses each edge in $G$ exactly once and therefore is an Eulerian circuit in $G$. Therefore it suffices to prove $H$, a connected graph with $m \geq 1$ edges, has an Eulerian Circuit.

We claim $\forall u \in V(H)$ $d(u) \geq 2$. Indeed, because $H$ has at least two vertices. Because $H$ is connected, each vertex in $H$ has degree at least 1. But vertices in $G$, hence and therefore vertices in $H$, have even degree. Therefore each vertex in $H$

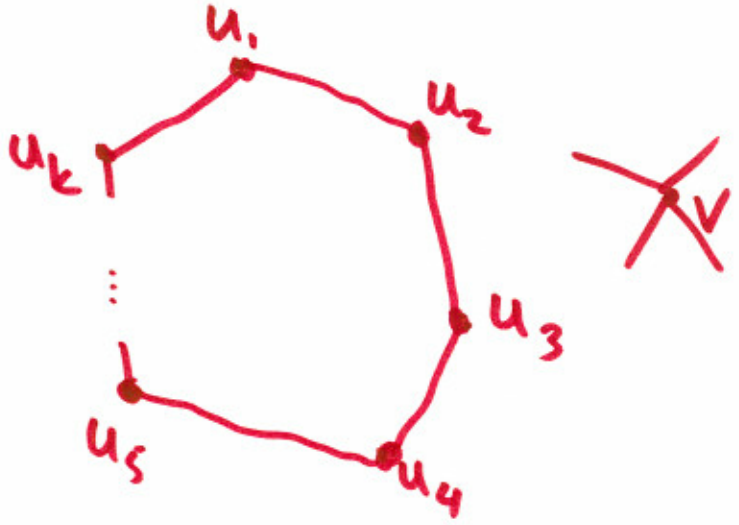
has even degree at least 2.

As we saw in Lecture 8, this means $H$ contains a cycle

$$C = u_1, u_2 \cdots u_k$$

Let $H'$ be the graph obtained from $H$ by deleting the edges in $C$.

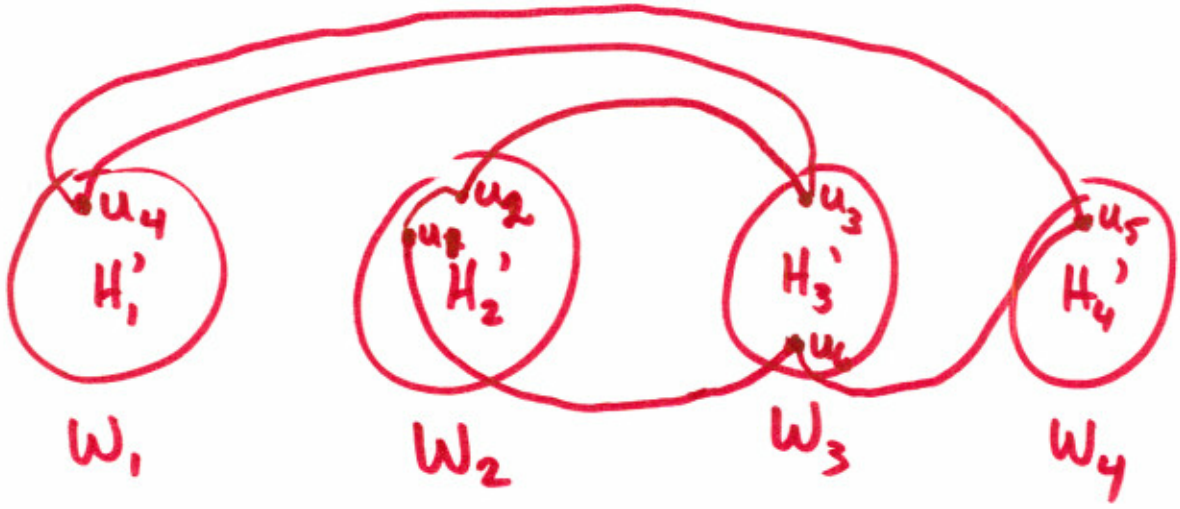
Note that each vertex in $H'$ has even degree.




Let $H'_1, H'_2, H'_3, \ldots, H'_r$ be the components of $H'$.

Note: Each $H_j'$ is a connected graph with strictly less than $m$ edges and each vertex has even degree.

Therefore, we may invoke the inductive hypothesis on each $H_j'$ to obtain an Eulerian Circuit $W_j$ of $H_j'$.




Because $H$ is connected, each component $H_j'$ of $H'$ must contain at least one vertex from $C$.

So, we may construct an Eulerian circuit $W$ in $H$ as follows. First, start out $u_1$

in $C$. We will have $W$ walk around $C$, except that each time $W$ enters a new component $H_j'$ that $W$ has not previously entered, $W$ will detour through $H_j'$ via $W_j$.

Because each $H_j'$ contains some vertex from $C$, $W$ detours through each of the components.

Because each $W_j$ traverses each edge in $H_j'$ exactly once and all edges in $H$ are either in $C$ or a component $H_j'$, $W$ is an Eulerian circuit. ∎

HW: When does $G$ contain an Eulerian trail?