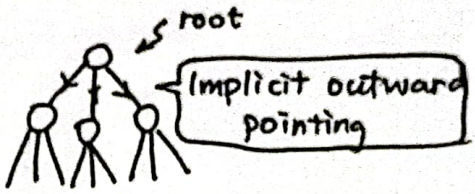
Jeff Erickson: Algorithms

§ 2.3 Dynamic Programming on Trees

1. Independent Set:

- Defn: a subset of verticies with no edges between them.

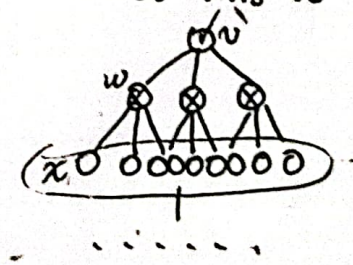
Given a tree T, rooted. Ask for Maximal Independent Set.



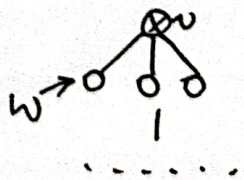
- Recursive defn. For any node $v \in T$, let $MIS(v)$ denote "Size of largest independent set in the subtree rooted at v"

a choosing problem — Select this to MIS? $1+\sum_{w \downarrow v} \sum_{x \downarrow w} MIS(x)$

$w \downarrow v :=$ w is a child of v



— or don't choose



$\sum_{w \downarrow v} MIS(w)$

$$MIS(v) = \max\left\{ \sum_{w \downarrow v} MIS(w),\ 1 + \sum_{w \downarrow v} \sum_{x \downarrow w} MIS(x) \right\}.$$

- Need to compute $MIS(r)$ ← root of the tree
- Memorized Structure.

The tree T itself!

- Order to consider subproblems

  Every vertex is visited before its parent

  (one possible option: post order traversal)

- Analysis: Each vertex contrib a const time $\Rightarrow O(n)$.

```
TREEMIS(v):
  skip v ← 0
  for each child w of v
     skip v ← skip v + TREEMIS(v)
  keep v ← 1
  for each grandchild x of v
     keep v ← keep v + [x.MIS]   ← Remembered
  v.MIS ← max{keep v, skip v}
  return v.MIS.
```

▷ Another appoarch

MISyes(v) := size of largest independent set of the subtree rooted at v, includes v.

MISno(v) := ... excludes ...

Recurrence Structure:

$$MISyes(v) = 1 + \sum_{w \downarrow v} MISno(w)$$

$$MISno(v) = \sum_{w \downarrow v} \max\{MISyes(w), MISno(w)\}.$$

```
TREEMIS2(v):
  v.MISno ← 0
  v.MISyes ← 1
  for each child w of v:
     v.MISno ← v.MISno + TREEMIS2(w)
     v.MISyes ← v.MISyes + w.MISno
  return max{v.MISyes, v.MISno}.
```