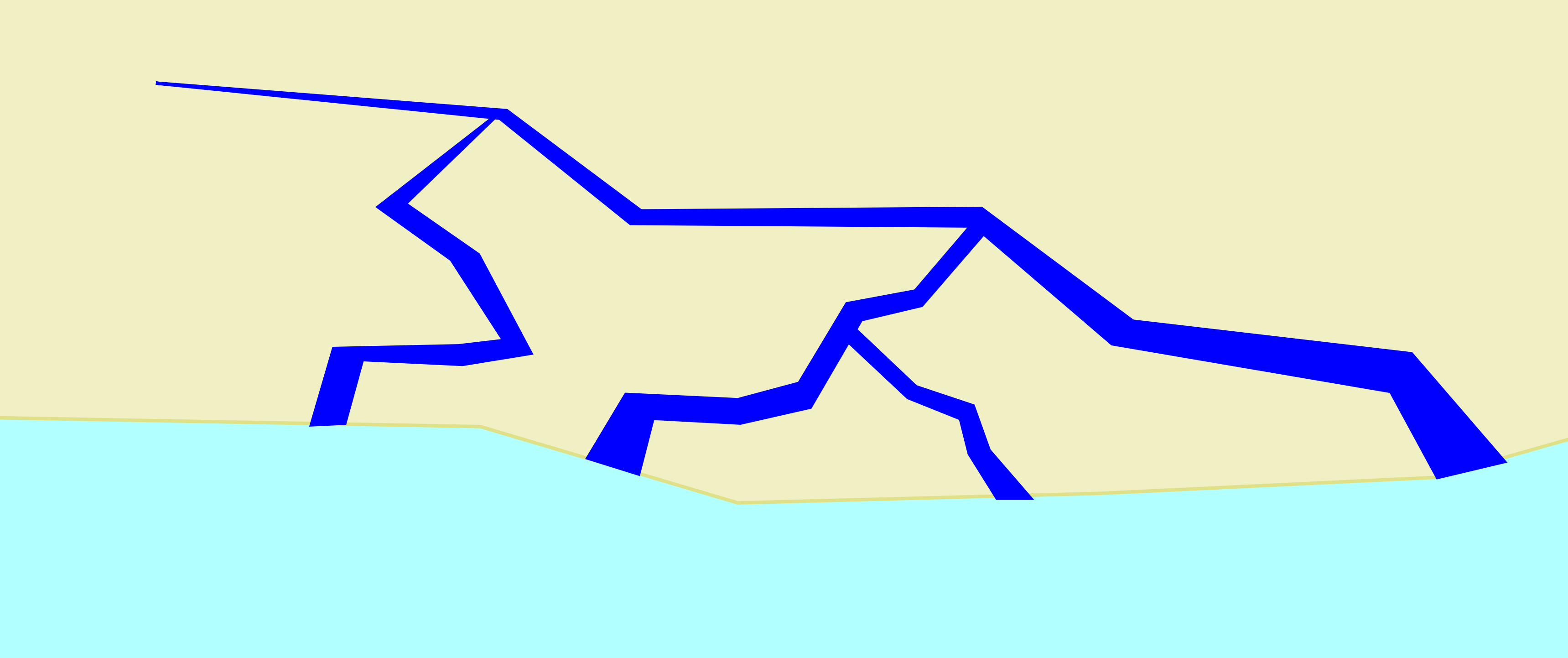
Winter 2016
COMP-250: Introduction to Computer Science

Lecture 17, March 15, 2016

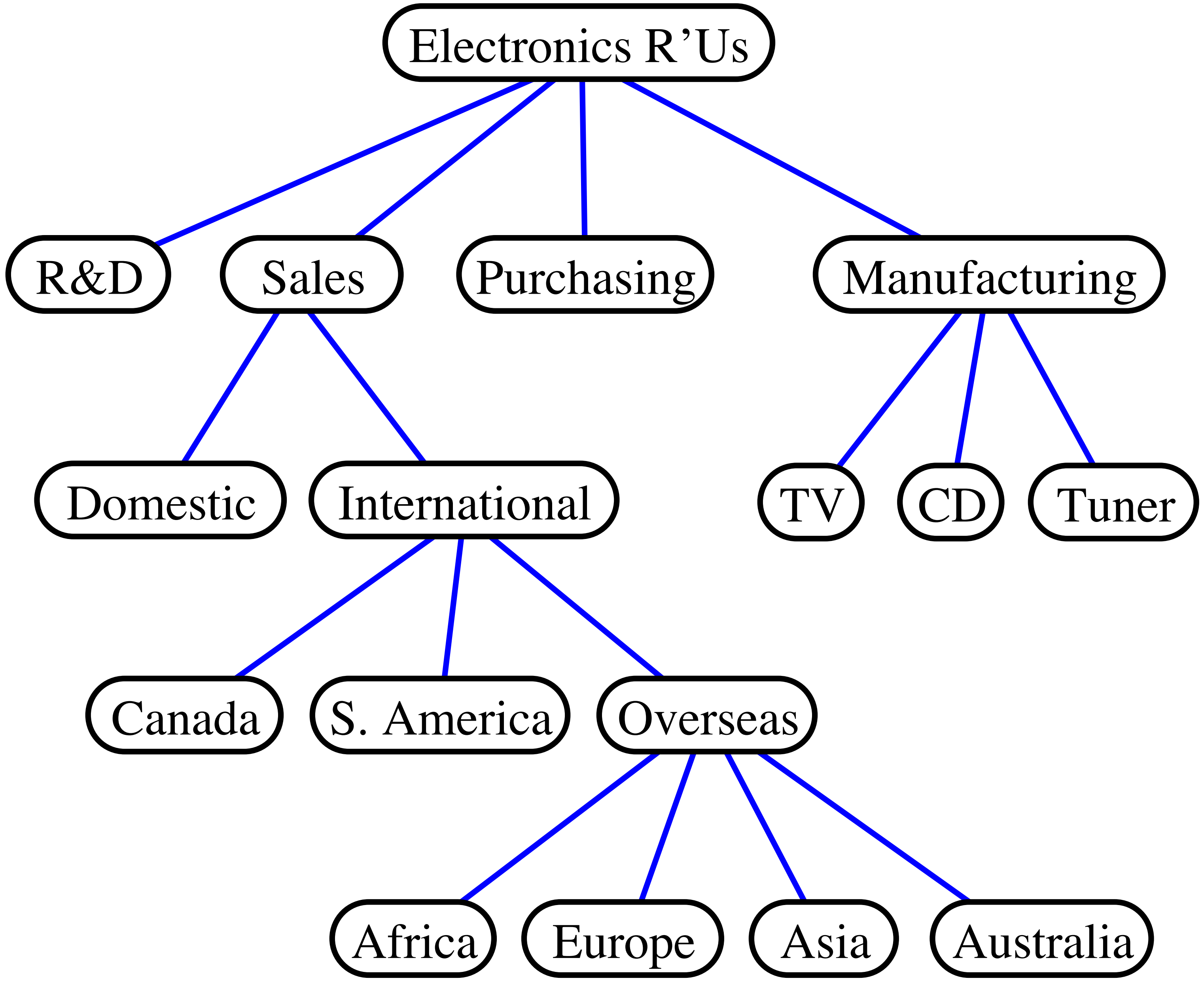
TREES

- trees
- binary trees
- traversals of trees
- template method pattern
- data structures for trees



Trees

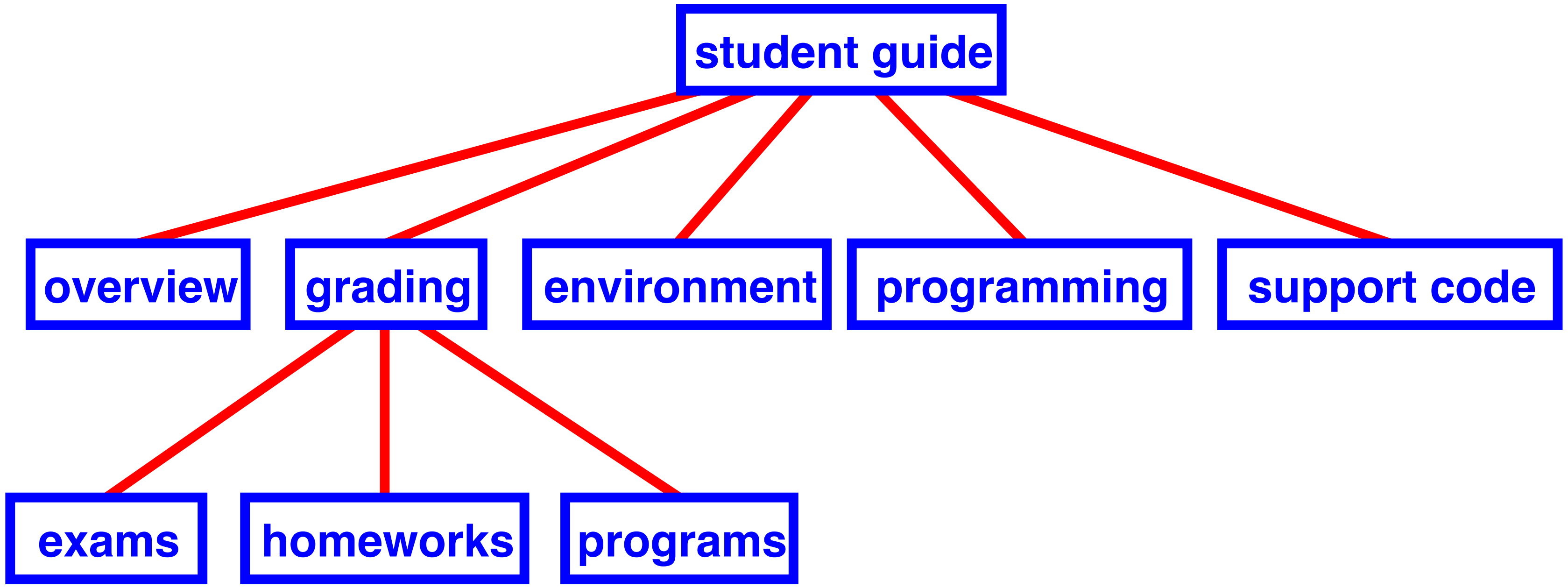
- a tree represents a hierarchy
  -organization structure of a corporation



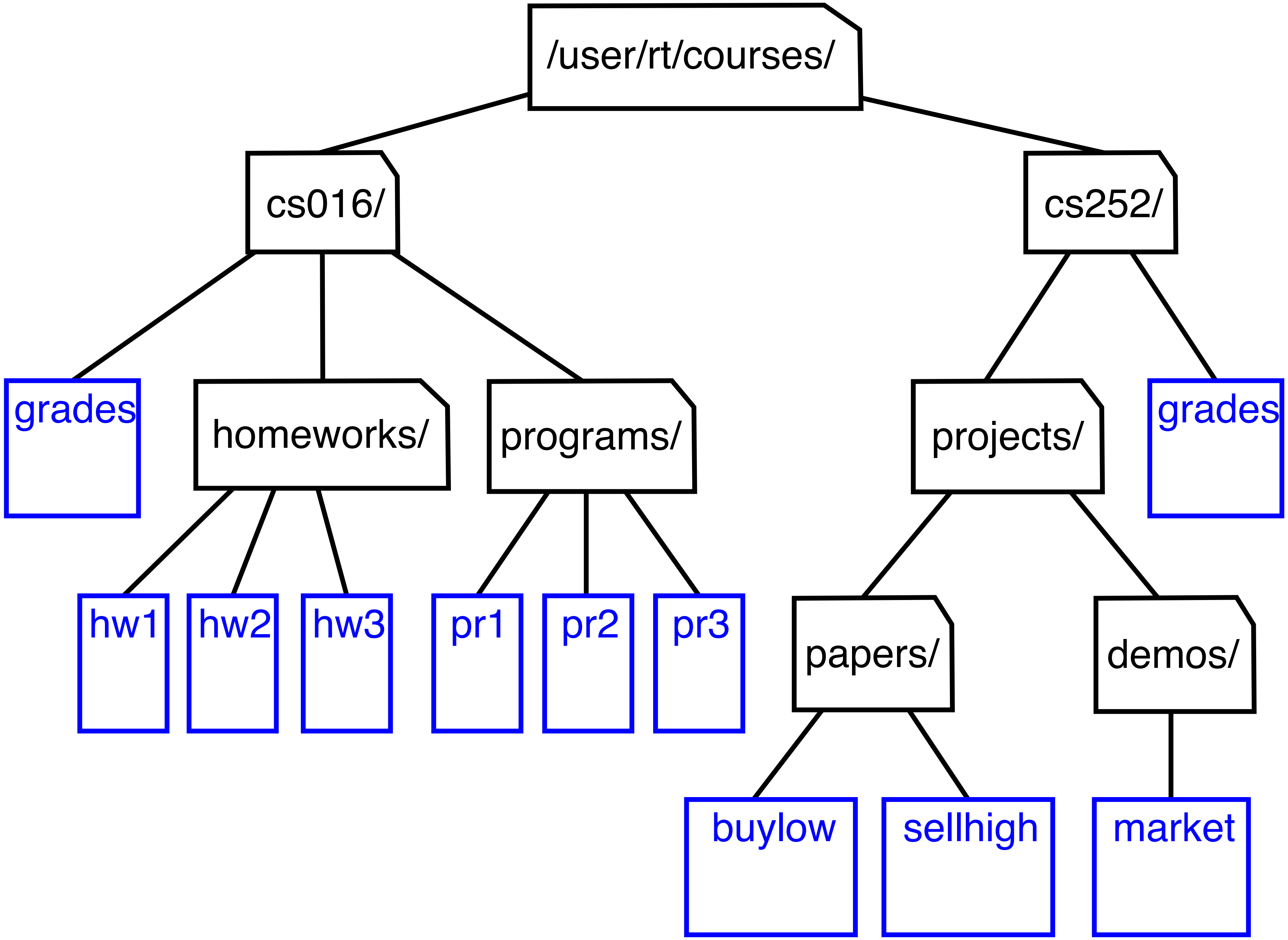
Trees

- a tree represents a hierarchy

  -table of contents of a book



- Unix or DOS/Windows file system



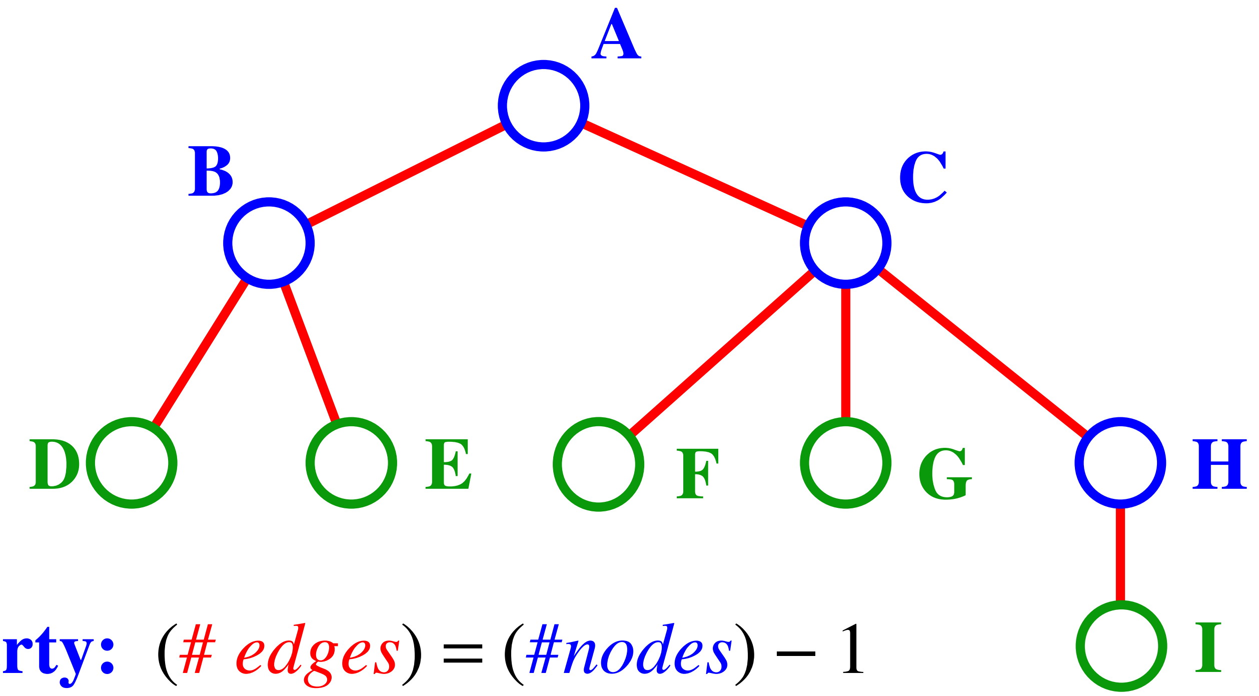
Terminology

- ***A*** is the ***root*** node.
- ***B*** is the ***parent*** of D and E.
- *C* is the ***sibling*** of B
- ***D*** and ***E*** are the ***children*** of B
- ***D, E, F, G***, *I* are ***external nodes***, **or** ***leaves***
- ***A, B, C, H*** are ***internal nodes***

•The ***depth*** (***level***) of ***E*** is ***2***

•The ***height*** of the tree is ***3***

•The ***degree*** of node ***B*** is ***2***



Property: (*# edges*) = (*#nodes*) – 1

Binary Trees

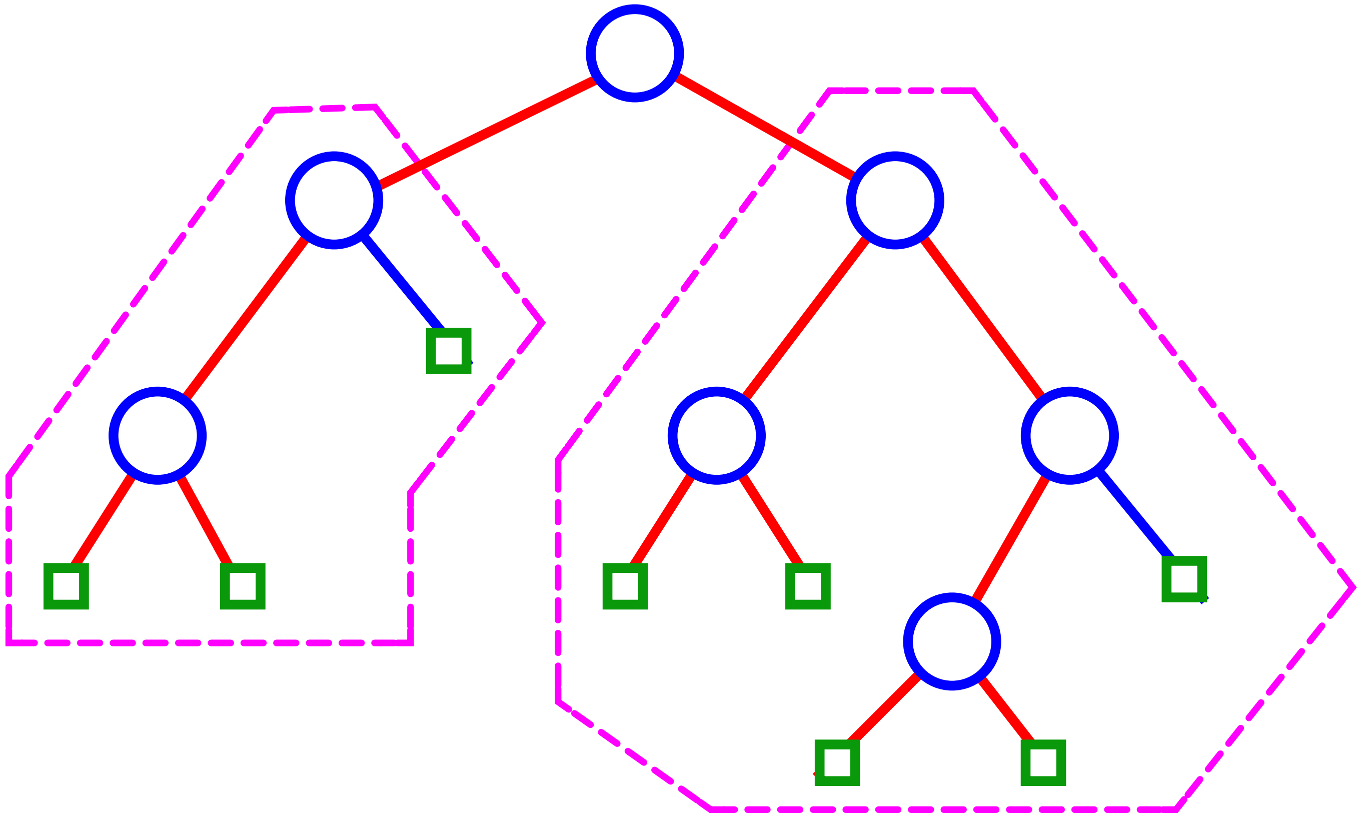
- ***Ordered tree:*** the children of each node are ordered.
- ***Binary tree:*** ordered tree with all internal nodes of ***degree 2***.

•Recursive definition of binary tree:

- A ***binary tree*** is either

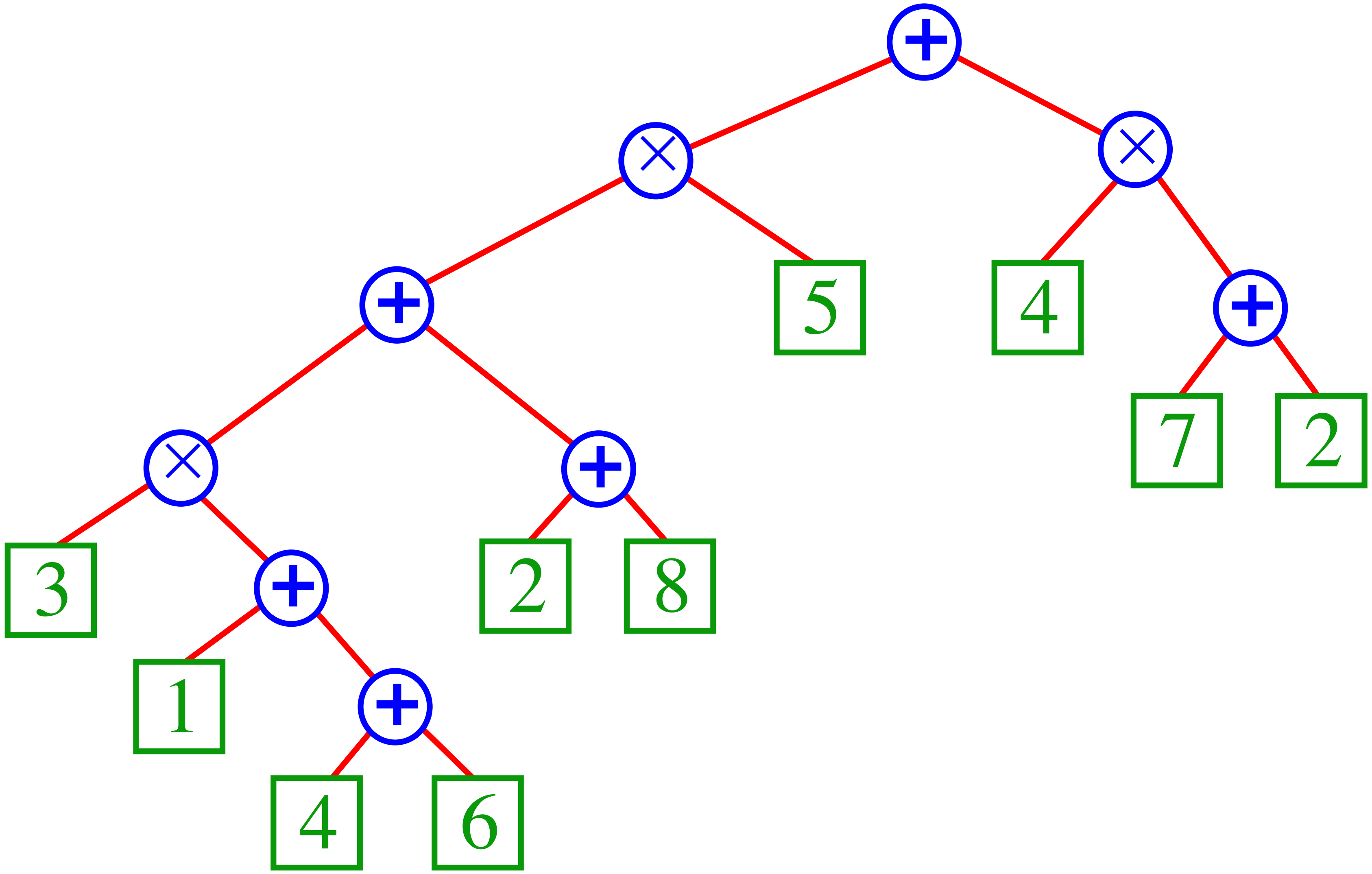
  - a n external node (leaf), **or**

  - a n internal node (the ***root***) and two binary trees (***left subtree*** and ***right subtree***)



Examples of Binary Trees

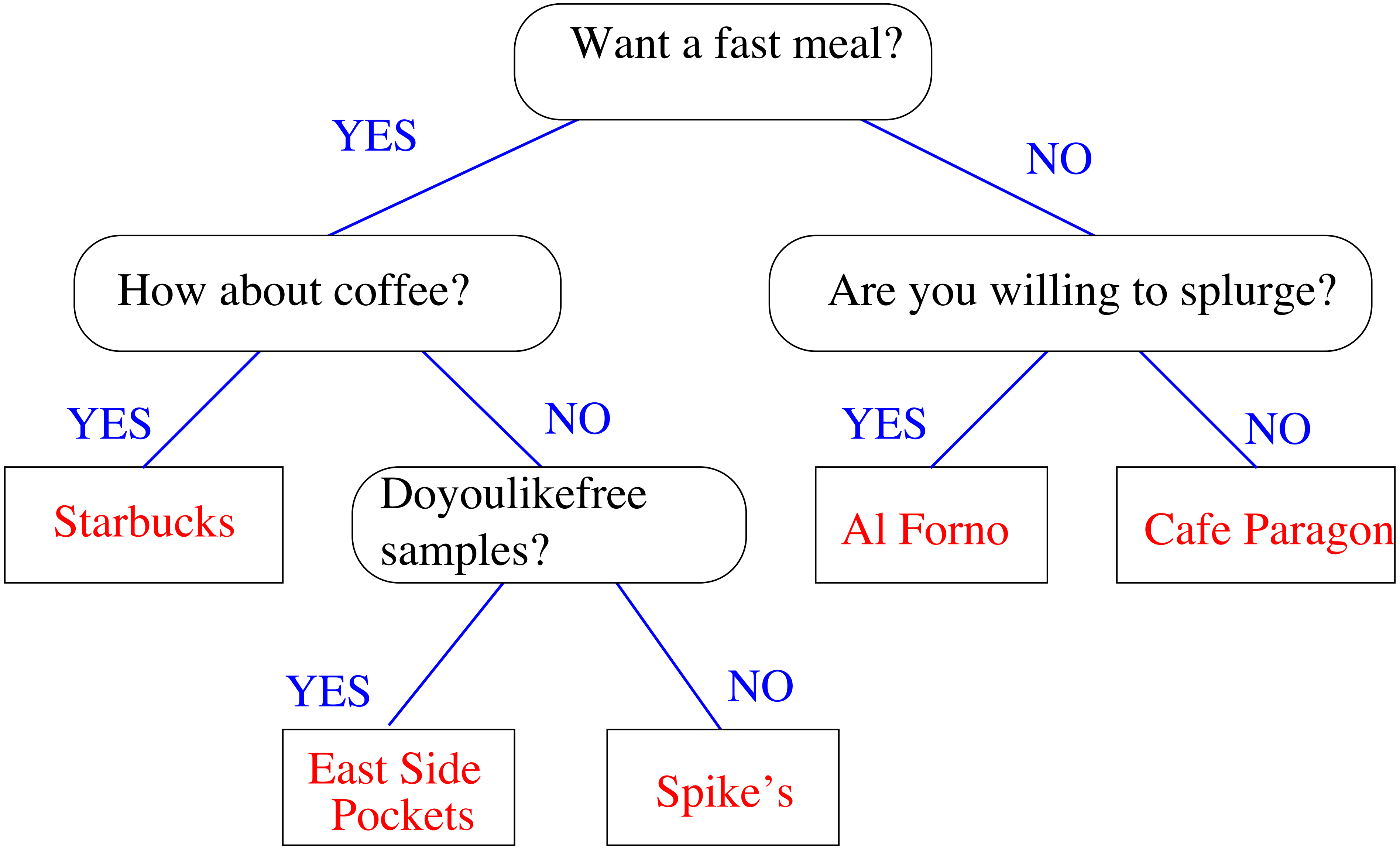
- arithmetic expression



((((3 × (1 + (4 + 6))) + (2 + 8)) × 5) + (4 × (7 + 2)))

Examples of Binary Trees

•decision trees



Want a fast meal?

YES

NO

How about coffee?

Are you willing to splurge?

YES

NO

YES

NO

Starbucks

Doyoulikefree samples?

Al Forno

Cafe Paragon

YES

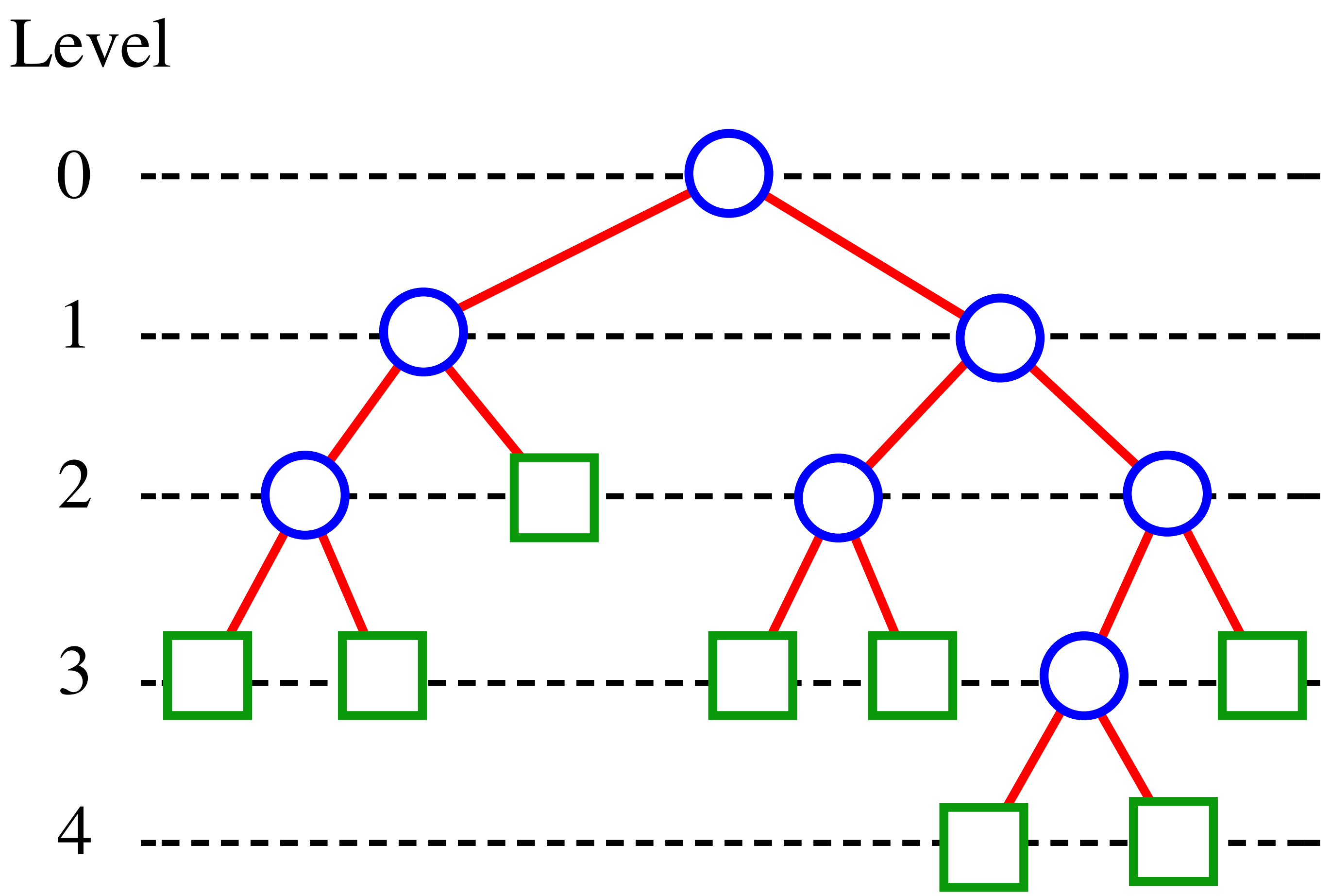
NO

East Side Pockets

Spike’s

Properties of Binary Trees

- (# external nodes) = (# internal nodes) + 1
- (# nodes at level i) $\leq$ 2^{i}
- (# external nodes) $\leq$ $2^{(\text{height})}$
- (height) $\geq$ $\log_2$ (# external nodes)
- (height) $\geq$ $\log_2$ (# nodes) – 1
- (height) $\leq$ (# internal nodes) = ((# nodes) – 1)/2



ADTs for Trees

•generic container methods

-size(), isEmpty(), elements()

•positional container methods

-positions(), swapElements(p,q), replaceElement(p,e)

•query methods

-isRoot(p), isInternal(p), isExternal(p)

•accessor methods

-root(), parent(p), children(p)

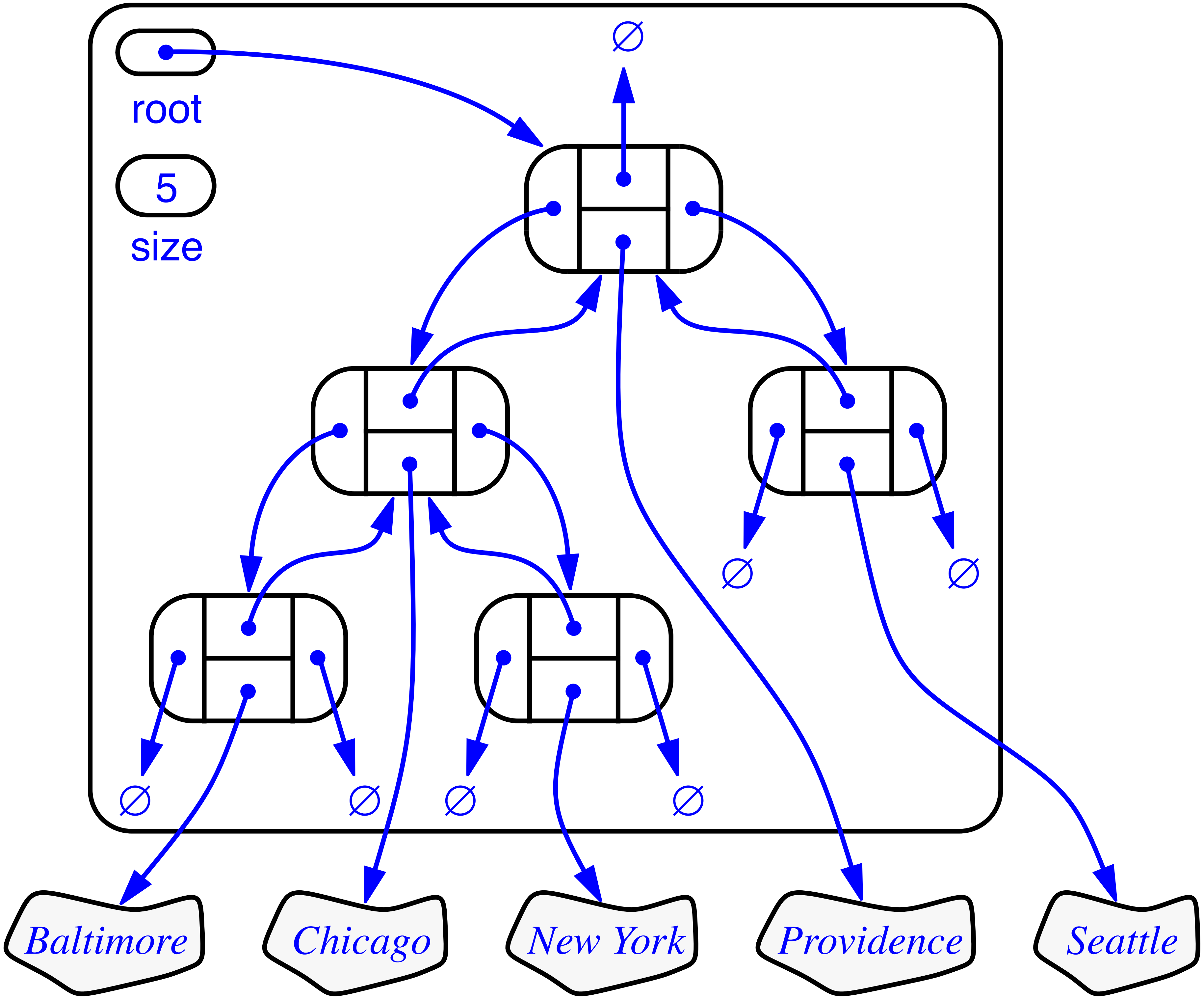
•update methods

-application specific

ADTs for Binary Trees

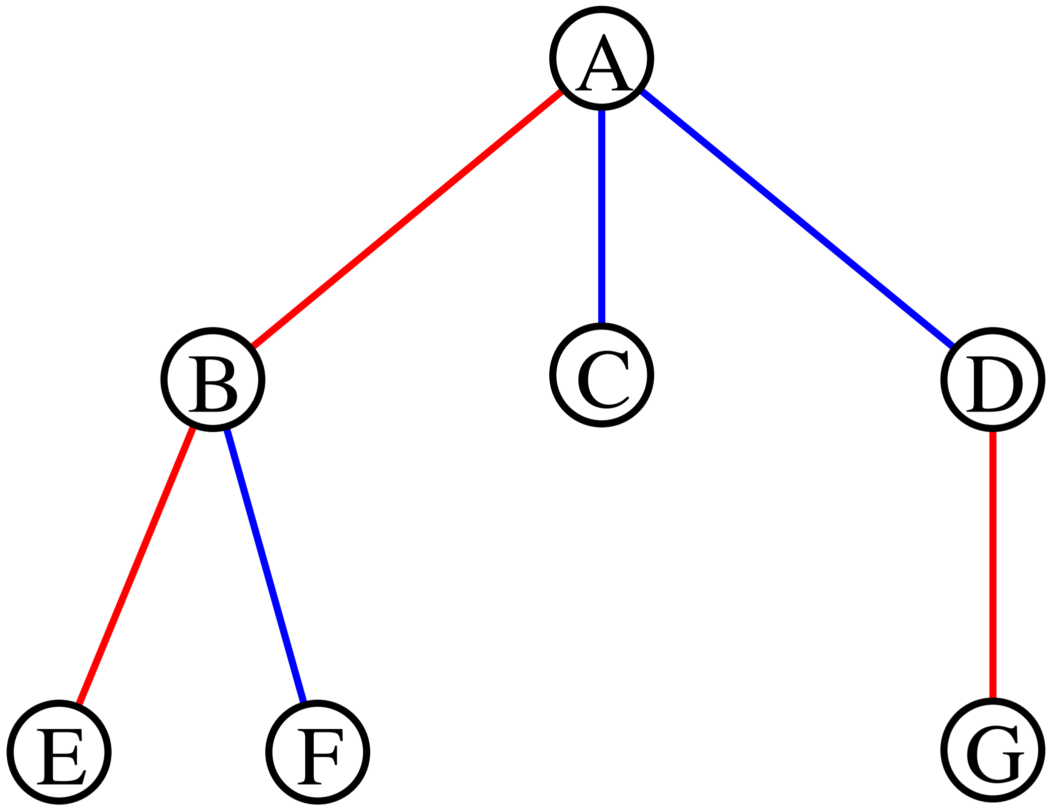
- accessor methods
  - leftChild(p), rightChild(p), sibling(p)
- update methods
  - expandExternal(p), removeAboveExternal(p)
  - other application specific methods

Linked Data Structure for Binary Trees



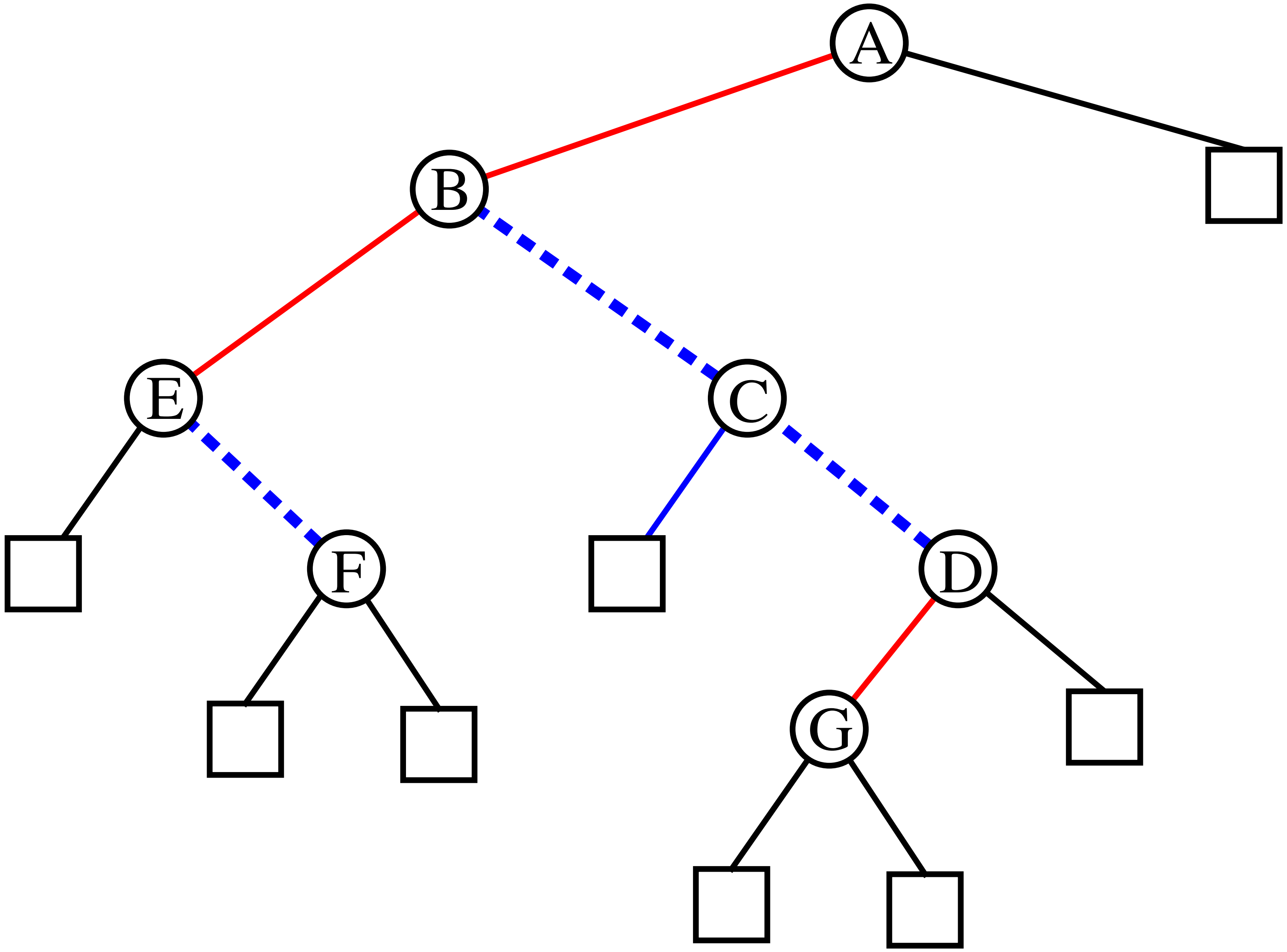
Representing General Trees

- tree T



Representing General Trees

- binary tree T' representing T



Winter 2016 COMP-250: Introduction to Computer Science

Lecture 17, March 15, 2016