How we represent trees in our Tree class

Empty tree

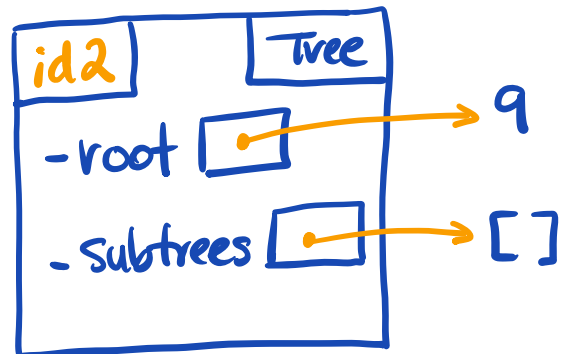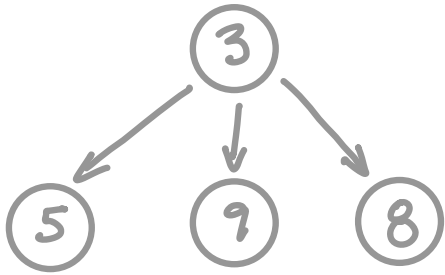| id1	Tree
- root [•]	
- subtrees [•] → [ ]	
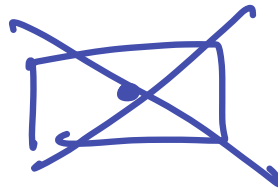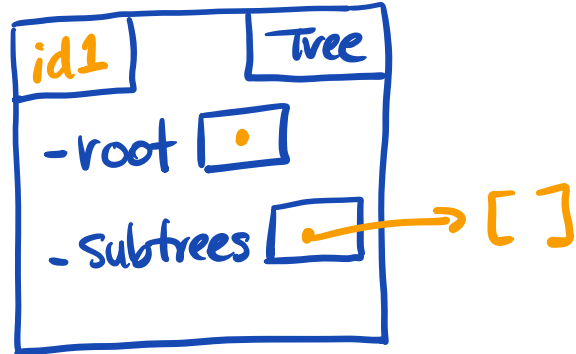
~~[•]~~ We did not choose to represent an empty tree as simply None.

Single value

(9)

id2	Tree
- root [•] → 9	
- subtrees [•] → [ ]	

Bigger

If we stored any meta-data (eg tree height), we'd store it in every Tree object
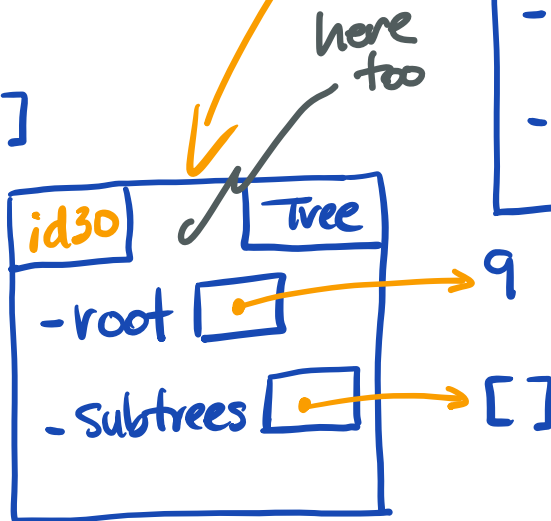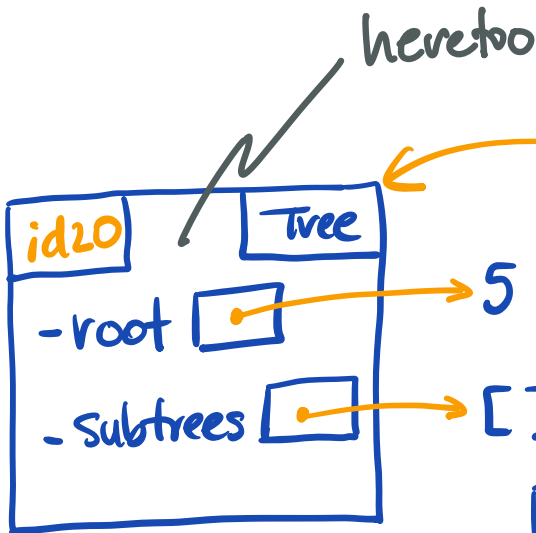
id 10   Tree
- root ☐ → 3
- Subtrees ☐ → | id20 | id30 | id40 |

heretoo

id20   Tree
- root ☐ → 5
- Subtrees ☐ → [ ]

here too

id30   Tree
- root ☐ → 9
- Subtrees ☐ → [ ]

here too
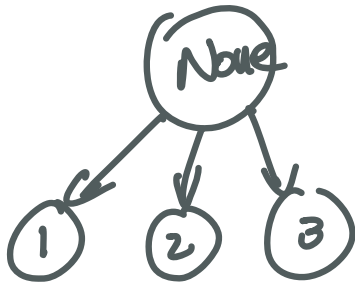
id40   Tree
- root ☐ → 8
- Subtrees ☐ → [ ]
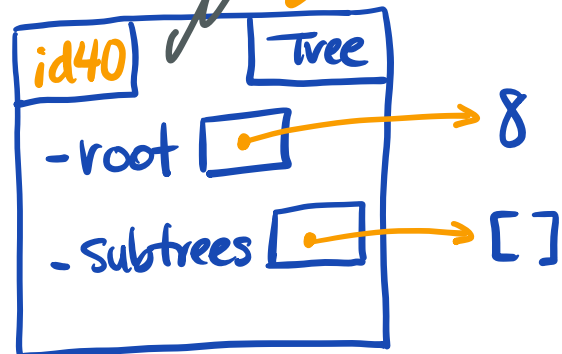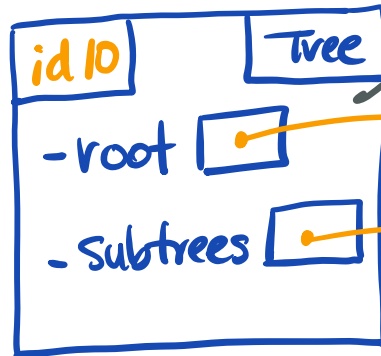
What do our Representation Invariants prohibit?

This is not
allowed ⟿

_root [●]
_subtrees [●]
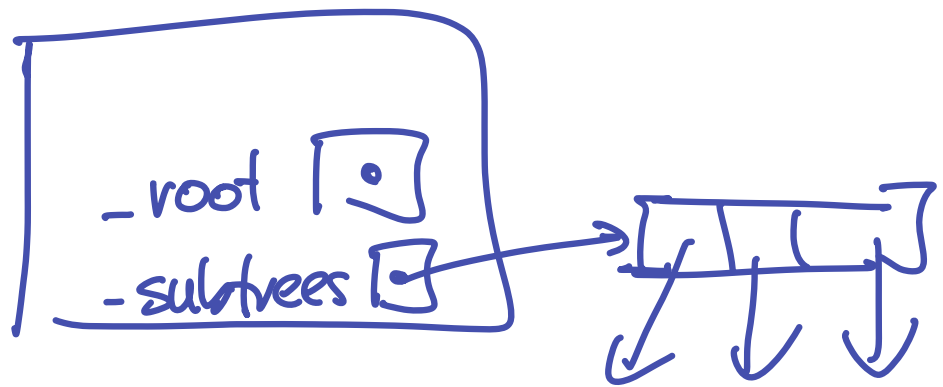
So we could never represent
a tree with None as an item inside it, such as
this tree

None
1 2 3

Another way to represent trees: with a separate node class.

3
├── **5**
├── **9**
└── **8**

id100 | **Tree**
_root | **id20**
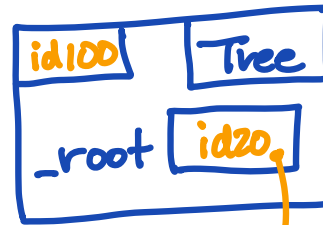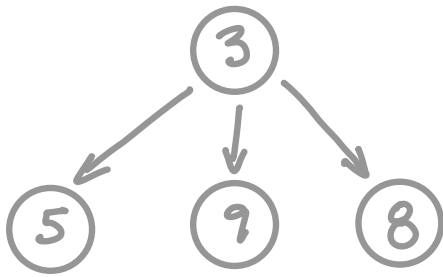
meta-data could go here, once

id10 | **Node**
-item → 3
-subtrees → **id20** | **id30** | **id40**

id20 | **Node**
-item → 5
-subtrees → [ ]

id30 | **Node**
-item → 8
-subtrees → [ ]

id40 | **Node**
-item → 9
-subtrees → [ ]