Part 1

ⓐ Node n1 = new Node("5"); // has to be a string

```
n1 ──┐
     data   next
    ┌──────┬──────┐
    │ "5"  │ null │
    └──────┴──────┘
```
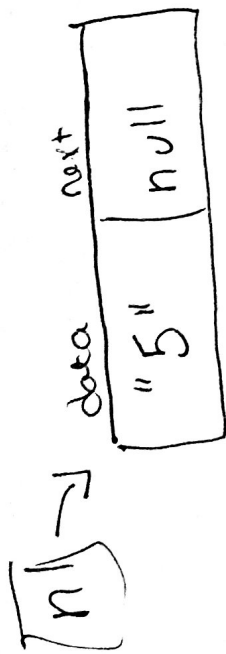
ⓑ Node n2 = new Node("10"); // has to be a string!!

```
n1 ──┐
     data   next
    ┌──────┬──────┐
    │ "5"  │ null │
    └──────┴──────┘

n2 ──┐
     data   next
    ┌──────┬──────┐
    │ "10" │ null │
    └──────┴──────┘
```

ⓒ n1.setNext(n2);

```
n1 ──┐
     data   next
    ┌──────┬──────┐
    │ "5"  │   ───┼──┐
    └──────┴──────┘  │
                     ↓
n2 ──┐               │
     data   next     │
    ┌──────┬──────┐ ←┘
    │ "10" │ null │
    └──────┴──────┘
```

ⓓ n1.setNext(new Node("15"));

```
n1 ──┐
     data   next
    ┌──────┬──────┐
    │ "5"  │   ───┼──┐
    └──────┴──────┘  │
                     ↓
n2 ──┐               │
     data   next     │
    ┌──────┬──────┐ ←┘
    │ "10" │   ───┼──┐
    └──────┴──────┘  │
                     ↓    no name node
                    ┌──────┬──────┐
                    │ "15" │ null │
                    └──────┴──────┘
```
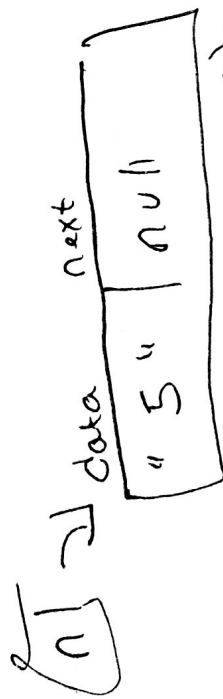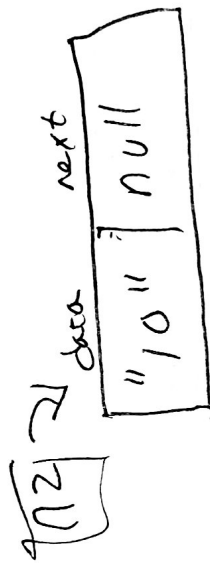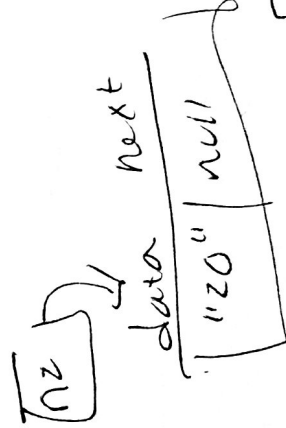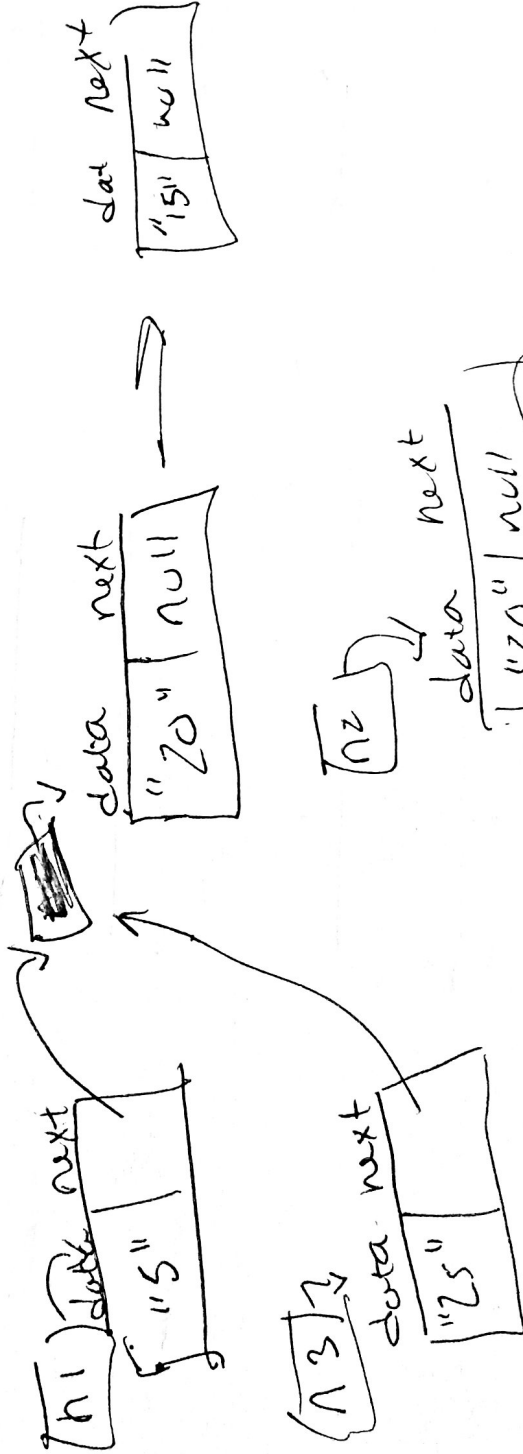
Part 1

© n2 = new Node ("20");

"no fone node",
from previous is
garbage collected
n2 points to new thing
Other wize se last
diagram

n1 [data | next] n2 [data | next]
   ["5" |]   → ["20" | null]

© Node n3 = new Node ("25", n2);

data   next
["20" | null]   →

[n2]
data   next
["20" | null]

n1 [data | next]
   ["5" |]

n3 [data | next]
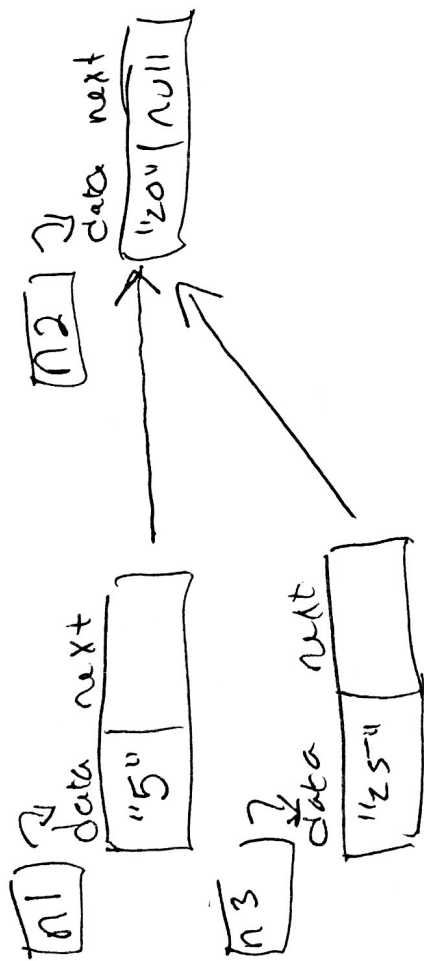   ["25" |]

dat   next
["15" | null]

I acknowleg after the fact that the arrows
should point to the contents not the
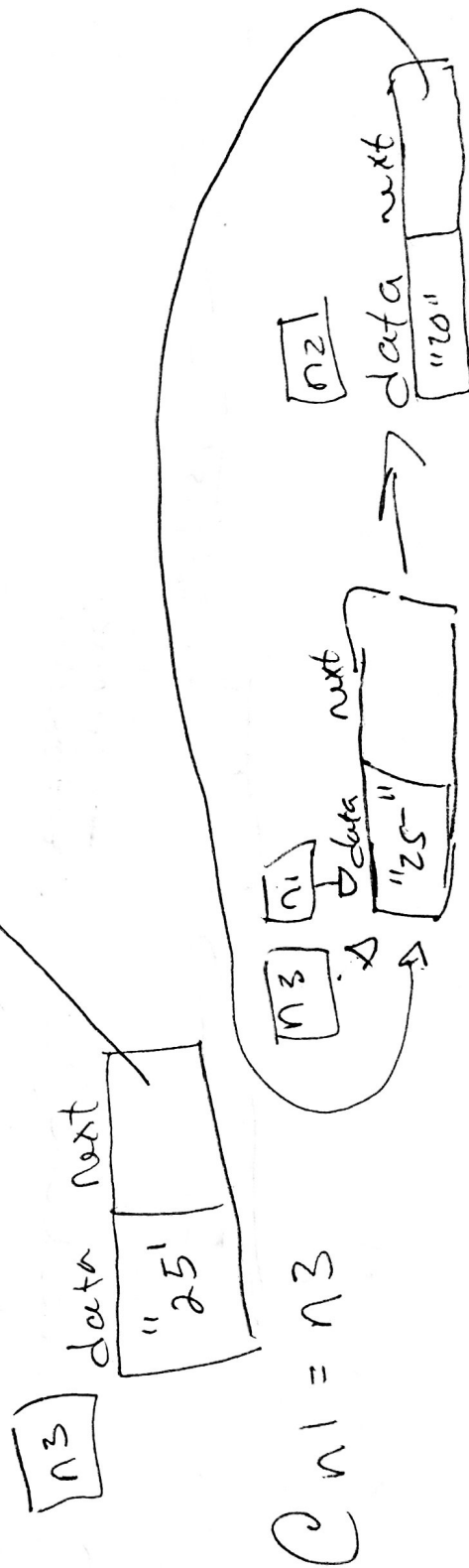var name. But dont want to write over cell's
                    □
                  → [|] (should be
□   have
[|]

Part 2

(a) Precondition

n1 → data next → "5"

n2 → data next → "20" | null

n3 → data next → "25"

(b) n3.setNext(n1);

n1 → data next → "5"

n2 → data next → "20"

n3 → data next → "25"

(c) n1 = n3

n2 → data next → "20"

n1 n3 → data next → "25"

Part 3

Given [P]
↓

5 | → 10 | → 15 | → 20 |

① Node newNode ("10", ~~Get nxt()~~ P. ~~nextNode()~~);

② Node aNode ("30");

③ ~~aNode~~ L1 aNode.setNext (P.getNext(s.getNext()));

   L2 P.getNext.setNext(aNode);