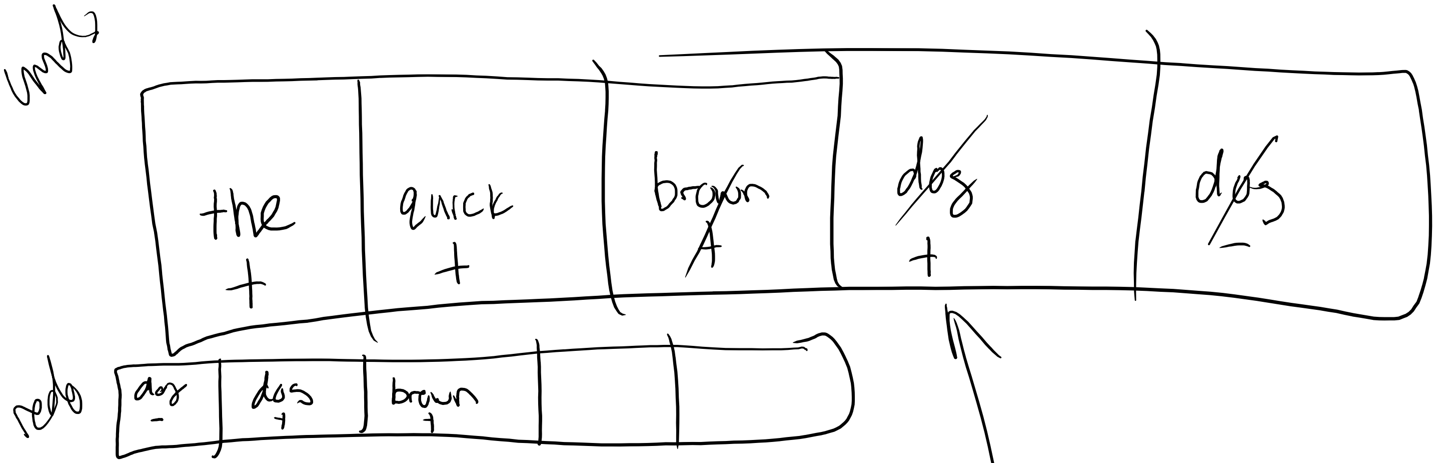
# 2018-01-30

Tuesday, January 30, 2018 9:00 AM



Question: how do we implement undo in our code?

Idea #1: use an array of sorts

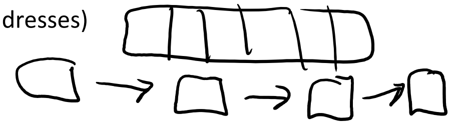
Theme: word processor. Granularity at the word level.

the quick ~~brown~~ ~~dog~~

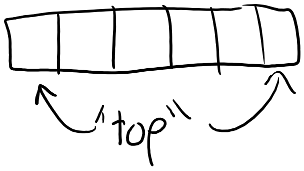
~~dog~~

## Stacks

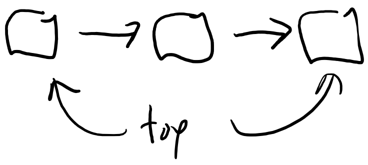
- New items go on "top"
- Items are removed from the "top"
- Classic analogy: stack of plates
- Stacks are used in many search algorithms
  - Preview: stacks are used in "depth first search" algorithms
- Stacks can be implemented using either vectors or linked lists
  - Vector: array (contiguous set of memory addresses)
  - Linked list: list of non-contiguous data.



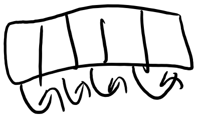
- When implementing a stack, we have a decision:
  - Where is the "top" of the stack?
- Vector



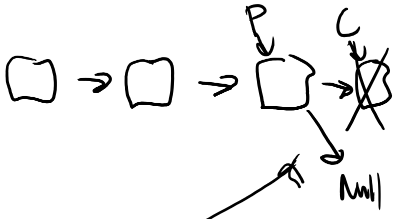
- Linked list



- Pseudocode for vector-based stack with "top" being element 0
  - "Push"
    - for(int i = size; i > 0; i--)
      - stack[i] = stack[i-1]
    - stack[0] = new item



        - stack[0] = new_item
    - "Pop"
        - result = stack[0]
        - for(int i = 0; i < size - 1; i++)
            - stack[i] = stack[i+1]
- Pseudocode for vector-based stack with "top" being last element
    - "Push"
        - stack[size] = new_item
        - size++
    - "Pop"
        - result = stack[size - 1]
        - size--
- Pseudocode for linked list stack with "top" being first element
    - "Push"
        - box = new box(new_item)
        - box->next = _front
        - _front = box
    - "Pop"
        - result = _front->value
        - old_front = _front
        - _front = _front->next
        - delete old_front
- Pseudocode for LL with stack "top" being last element
    - Assumption: no "end" pointer
    - "Push"
        - current = _front
        - while current->next != nullptr:
            - Current = current->next
        - box = new box(new_item)
        - Current->next = box
    - "Pop"
        - current = _front
        - Prev = current
        - While current->next != nullptr:
            - Prev = current
            - Current = current->next
        - Value = current->value
        - Delete current
        - Prev->next = nullptr



    - If we have an "end" pointer, code looks nearly identical to code for when "top" is front