30-01-2018 – COMP15 Notes

Outline

Reading: Shaffer 4.12, Weiss 3.2

1. Run-Time Complexity
2. Linked List Operations
3. LL v. AL

- getting value in array takes one step because an AL is contiguous

I. Runtime Complexity

- measure efficiency of an algorithm
- size of input (n)
- no. of steps relative to n as n goes arbitrarily large
- one step = constant runtime
- O(1) - constant
- O(lgn) - logarithmic
- O(n) - linear
- O(n^k) - polynomial

II. Linked List Operations

- LL offers:
  - dynamic memory
  - no expand (copy)
  - length == capacity
- every element is a NodeType in a linked list

  ```
  struct NodeType
  {
      Planet info;
      NodeType *next;
  }
  ```

  - head always points to first element
  - last element’s next always points to null
  - not contiguous in memory
  - allocate NodeTypes one at a time

III. LL v. AL

    - is_full O(1) v. O(1)
    - find O(n) v. O(n)
    - remove O(n) v. O(1)
    - make_empty O(1) v. O(n)