Andrew Plum

Cs 121

2/28/22

Week 8 Notes

* Pointers and objects
  + cout << (n.next)->data << endl;
  + cout << ((n.next)->next)->data << endl;
    - Need parentheses because of order of operations
      * I don’t know the order
* Given an unsorted list, how long on average will it take to find a name?
  + n/2 = O(n)
* Given an sorted list, how long on average will it take to find a name?
  + O(Log2(n))
    - This means jump to the middle, then jump in the middle of what’s left, and repeat until you find it
      * Every time you jump you throw away half of the list
      * This is where log is used for and it is way more significant mathematically
* If doing a insert\_sorted() for a link list then
  + When you insert, you need to compare the number you are inserting with the number the pointer is pointing to and the number after the one the pointer with your number is pointing to
  + Special Cases
    - Check first case head points to
    - Check Empty case
* Insert\_sorted(int x){

If(next->data > x || next == NULL){

Node \*p = new node();

p->data = x;

p->next = next;

next = p;

return;

}

Next->insert(x);

}

Void::link\_list(x){

If(front == NULL || front->data > x){

Node\* p = new node();

p->data = x;

p->next = front;

p->next = front;

front = p;

if(back == NULL){

back = p;

} else {

front->insert(x);

}

}

}

* Trees
  + Data structure
  + Can be binary, trinary, end to end
  + Leaves and root
    - Leaves are on the ends
  + Parents and child
  + Height of the tree
    - # of generations
* N items
  + Algorithm 1 which takes f(n) steps
  + Algorithm 2 which takes g(n) steps
    - If lim n->infinity (f(n)/g(n)) = 0
      * f(n) is faster than g(n)
    - If lim n->infinity (f(n)/g(n)) = constant
      * f(n) is equal to g(n)
    - If lim n->infinity (f(n)/g(n)) = infinity
      * f(n) is slower than g(n)
    - Assume all steps are equal