

# CS280 – Sorting Algorithms, Recursion

## January 25, 2016

<http://azrael.digipen.edu/~mmead/www/Courses/CS280/Sorting-1.html>

### *Misc. Things from Start of Class*

- Memory manager is graded and in return folder
- In the sample code for the Blist assignment, this line "const throw (Blist exception)" is now deprecated
- Use a binary search in our 'find' function
- Don't use the ternary operator ( ? : ) on a test
  - Also don't do redundant computations
  - e.g. `if(function(x,y) {function (x,y);}`
  - Store `function(x,y)` result in a variable
- We won't go in depth, but there are these useful things called expression trees
  - How the compiler evaluates recursive calls

### *Sorting*

- **Bubble sort** is one of the worst sorts
  - It is called a bubble sort because the large elements are slowly (swapped all the way to the top) 'bubbled' to the proper spot in the array
  - $O(N^2)$
  - $\Omega(0)$
  - We aren't using random access, so this works for linked lists
- **Insertion sort**
  - $\Omega(N)$  – Comparisons
  - $\Omega(0)$  – Shifts
  - $O(N^2)$  - Shifts

- $O(N^2)$  – Comparisons
- Average case is somewhere between  $N$  and  $N^2$
- We could use binary search to find the insertion point
- **Selection Sort**
  - Keep scanning through, finding smallest value and putting it in the correct spot
  - $\Omega(0)$  – Swaps
  - $\Omega(N^2)$  – Comparisons
  - $O(N)$  – Swaps
  - $O(N^2)$  – Compares
  - Would work for linked lists
- **Misc. Things on Sorting**
  - Sometimes the "almost best" algorithm is better than the best algorithm if it makes it easier and faster to implement and maintain
  - Different algorithms for different types of data, none is necessarily the best for everything
  - When given a question on the exam, we will get functions.
    - e.g.  $2x^2 + 8x + 100 \rightarrow O(x^2)$
  - All data structures will have some sort of analysis
  - Selection sort could be made twice as fast by doing both the largest and smallest values at the same time
  - *"It is a fact, all browsers just suck."*
  - Next homework is all about recursion

## ***Recursion***

<http://azrael.digipen.edu/~mmead/www/Courses/CS280/Recursion2.html>

- Want to use the question in the answer
  - e.g. "How do I quicksort?" --> "Partition the data, then quicksort the left and quicksort the right."
- If you can write the problem/solution in English first, you can write the code
- There are entire branches of computer science devoted to proving code does specific things
- Tail Recursion
  - Last statement is the recursive call
- You can get about 10,000 recursive calls deep before the stack overflows with variables and return addresses to the previous recursive calls
- All compilers today are optimized to detect recursive calls

*Optimizing a recursive call with a goto*

```
int Value = 5;

void PrintDown2()
{
    top:
    if (Value > 0)
    {
        cout << Value << endl;
        Value--;
        goto top;
    }
}
```

*Example of a quicksort, using the middle as the pivot point*



