Week 7
This Week

- Notepad++
- Mercurial
- Valgrind
- Bitwise Operators
- Data Structures (Stacks, Queues, Hash Tables, Binary Search Trees, Tries)
• Tried emacs but still hate the terminal? 😞
• Work in a GUI environment connected directly to the Cloud!
• This is better than WinSCP and a text editor. You don’t want to do that.
Mercurial

- RCS – Revision Control System

- Better than doing this:
Valgrind

• Pronunciation: val-grinned
• For best results:
   – valgrind –v –leak-check=full <program_name>
• Gives a report on status of memory allocated.
Bitwise Operators

& - AND
1100
&1010
1000

^ - XOR
1010
^1100
0110

| - OR
0011
|1010
1011

~ - NOT
~(1010) = 0101
Bitwise Encryption

One-time pad:
My string: 10001110
Encryption Key: 10011001
XOR Encrypted: 00010111
XOR Decrypted: 10001110
int swap(int* x, int* y) {
    *x ^= *y;
    *y ^= *x;
    *x ^= *y;
}

<table>
<thead>
<tr>
<th>X - 1001</th>
<th>Y - 1100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0101</td>
<td>1100</td>
</tr>
<tr>
<td>0101</td>
<td>1001</td>
</tr>
<tr>
<td>1100</td>
<td>1001</td>
</tr>
</tbody>
</table>
Data Structures

- Stacks
- Queues
- Hash Tables
- Binary Search Tree
- Tries
Stacks

- First in, last out data structure.
- Can ‘pop’ or ‘push’ things to the top of the stack.
Queues

• First in, first out data structure.

• “Insert” and “Remove” operations.
Hash Tables

• Consists of an array and a hash function.

• Hash function maps input to an index in the associated array.

• Allows us to check whether something is contained in a data structure without checking through the entire thing.
Hash Tables

Good Hash Functions are:
• Deterministic
• Well-distributed

```c
int xkcd_hash(char* word)
{
    return 4;
}
```

THIS IS BAD
Binary Search Tree

• Trees consist of ‘branches’.

```
struct branch
{
    struct branch* left;
    int val;
    struct branch* right;
}
```
Binary Search Tree

BST is such that:
1) Left subtree of each node contains only lesser nodes.
2) Right subtree of each node contains only greater nodes.
3) Left and right subtrees of each node are also binary search trees.
Tries

- Tree of Arrays
- Fast Lookup, High Memory Use

```c
struct trie_node
{
    struct trie_node* array[N];
    bool checkbox;
}
```
First two elements are in an array. First represents the letter ‘a’. Second represents the letter ‘b’. Checkbox indicates whether what we’ve looked at so far is in the data structure.

“a”, “bb” are in this structure.
THE BIG BOARD