From Trees to Graphs to Tries

- Huffman tree is really a trie
  - information retrieval
  - good for storing prefix trees
  - branching factor = alphabet

- Trie can be used to store words, prefix stored only once

```c
struct Trie
{
    Vector<Trie *> links;
    bool isWord;
};
```

- Print Trie? Add a word?
Huff Stuff

- What are key components?
- What is the pseudo-eof char?
  - How are bits read?
  - How are bits written
- Why the caveat on using chars?
- What about templates?
- What about Makefiles?
- What about make depend?
- What about unhuff reading?
- What about the unhuff header?
- What about ???

"character" counts

Huff tree/trie

Compressed header

21 32 41 0 51
010101010101010101010101
non-zeros? Traversal?