

Ethan Henry
CSE 13S
Professor Long
3/7/2020

DESIGN.pdf

PSEUDOCODE:

Trie:

Trie_node_create:

```
New = malloc();  
new->code = code;
```

Trie_create:

```
New = malloc();  
New_code = EMPTY_CODE;
```

Trie_step:

```
If (n->children[sym]):  
    Return n->children[sym];  
Else:  
    Return NULL;
```

Io:

```
UInt8_t buffer[4096];
```

Write_header:

```
write(outfile, header, sizeof(FileHeader));
```

Read_header:

```
read(infile, header, sizeof(FileHeader));
```

Read_sym:

```
Bool end = false;  
If (!end):  
    read(infile, buffer, 4096);  
    *syms = buffer[index];
```

```
Index++;
```

Buffer_pair:

```
for(index = 0; index < bitlen; index++):  
    Pos = index % bitlen;  
    Flag = 1 << pos  
    If ((code & flag) != 0):  
        pairs[pairs_index/8] |= (1 << (pairs_index % 8));  
    Pairs_index++;  
  
for(index = 0; index < 8; index++):  
    Pos = index % 8;  
    Flag = 1 << pos;  
    If ((sym & flag) != 0):  
        pairs[pairs_index/8] |= (1 << (pairs_index % 8));  
    Pairs_index++;
```

Flush_pair:

```
write(outfile, buffer, pairs_index)
```

Read_pairs:

```
If (index == 0) // It will always be at the start of the program  
    Read  
Get_bit from BitVector Code  
If (buffer == full):  
    Read  
    Index = 0;
```

Buffer_word:

```
Just putting each character in the buffer. All are 8 bits so its just  
For (index < word len):  
    Buffer[word_index] = word sym[index]  
    Word_index++
```

Flush_words:

```
write(outfile, buffer, words_index)
```

Word_create:

```
    Word new = malloc;  
    new->syms = malloc;  
    new->len = len
```

Word_append_sym:

```
    Word *new = word_create(w->syms, w->len + 1);  
    new->syms[w->len] = sym;
```

Word_delete:

```
    free(w->syms)  
    free(w);
```

Wt_delete:

```
    For (index = 0; index < MAX_CODE):  
        word_delete(w[index])  
    free(w);
```

Wt_create:

```
    WordTable new = calloc(MAX_CODE)  
    Word empty = word_create(NULL, 0)  
    new[EMPTY_CODE] = empty;
```

Wt_reset:

```
    For (index = 2; index < MAXCODE):  
        word_delete(wt[index])
```