

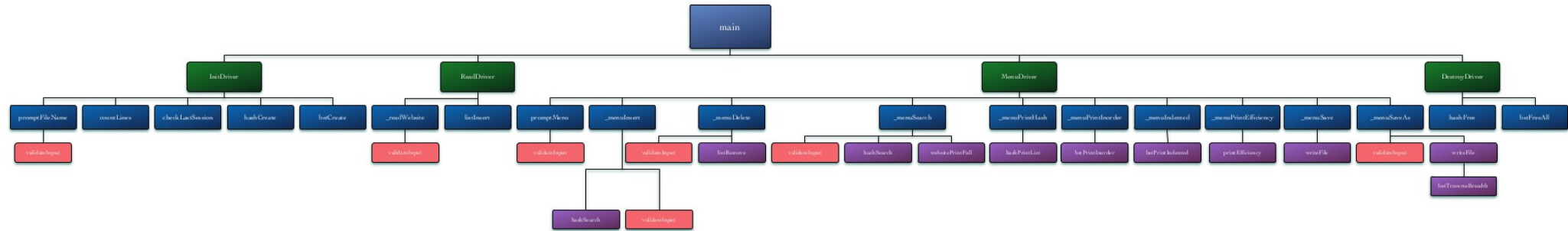
Website Catalog

Group 6:
Cory Sherman
Gon Kim
Chris Huang

Project Description

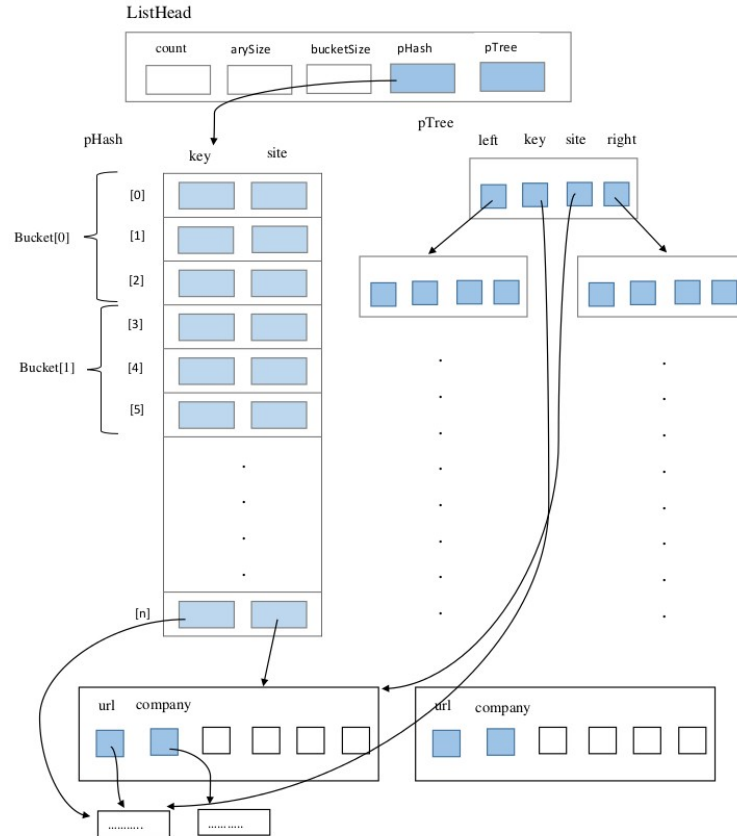
- Catalog top web domains and data about them
- Data parsed from Alexa and WebsiteOutlook
- Top 500 domains, 100 Unicode, 29 incomplete
- BST breadth-first traversal
- Hash buckets

Structure Chart



- Breadth-first traversal

Data Structure Diagram



- Collision resolution: buckets
 - Size: 3
 - One array for entire hash table
- Needs a lot of memory
- Cannot hold more than 3 synonyms for any given value.

Responsibilities: Cory Sherman

- Team Leader
 - Coordinate group
 - GitHub repository
 - Header File
 - Documentation
 - Design
 - Debugging and General Help
 - This Presentation
- Input Data
 - Wrote script to download and parse data about top 500 sites from Alexa.com and WebsiteOutlook.com
- BST Manager
 - Insert
 - Delete
 - Destroy
 - Breadth-first Traversal to File
 - Print Inorder
 - Print Indented
- Linux Build Scripts / makefile

Responsibilities: Gon Kim

- I/O and Input Validation
- Menu Manager
- Main
- Help with Design
- Help with Documentation
- Structure Chart
- OSX Build Scripts / makefile

Responsibilities: Chris Huang

- Hash Manager
 - Create
 - Find Prime
 - Insert
 - Search
 - Remove
 - Print Efficiency
 - Print in Hash Order
- Data Structure Diagram
- Visual Studio Project

Hash Function

- Used by the Java library to hash strings.

$$h = 31^{(n-1)}c[0] + 31^{(n-2)}c[1] + \dots + c[n-1]$$

```
int hash(const char *key)
{
    int h = 0;
    const char *w;

    for (w = key; *w != '\0'; w++)
    {
        h = 31 * h + *w;
    }
    return h;
}
```


Demonstration