Discussion 4

Haochen Yang

yhcyang@ucdavis.edu

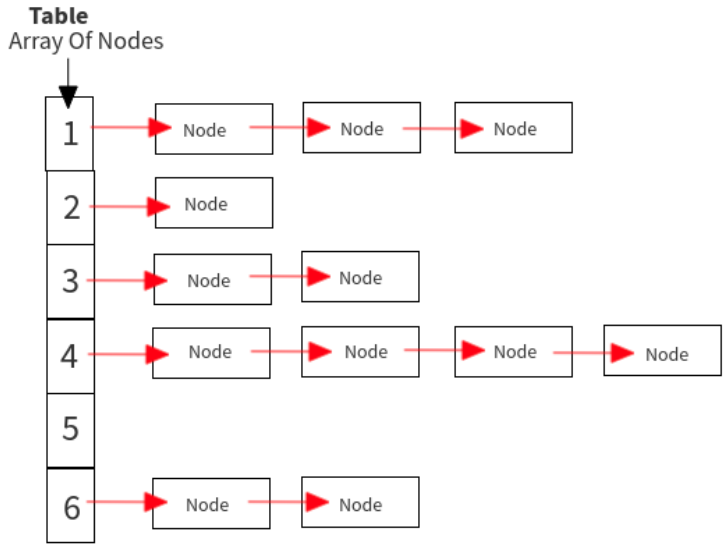
Announcement in Canvas

Roadmap

1. Quick Review
2. Implementation of Hash functions
3. Method to solve Hash collisions
4. Stack and Heap
5. FAQ for hw2

Lecture content(Quick review):

Hash table

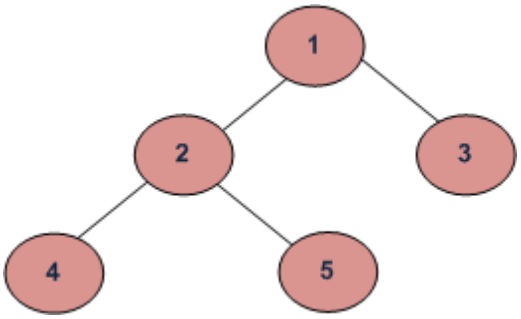


Java Hashmap internal Implementation

- **Characteristic**: Key, value
- **Hash function**: Below
- **Complexity for searching**: $O(1)$ → Also Insert / Delete
- **Hash collision:** Below

Binary tree:

- **Characteristic**: **Two** child, each child is a binary tree or nullptr
- **Traversal or walk**: preorder, inorder, postorder



inorder(left, root, right): 4 → 2 → 5 → 1 → 3

preorder(root, left, right): 1 → 2 → 4 → 5 → 3

postorder(left, right, root): 4 → 5 → 2 → 3 → 1

Hash functions

Def: Domain → co-domain

A good hash function would minimize the chance that such variants hash to the same slot.

The division method:

$$h(k) = k \mod m$$

k → input key, m → number of the slots in hash table

Rule of thumb: Try to find a prime which is not too close to an exact power of 2

Disadvantage: Much rely on the choice of m

- if $m = 4 = 0100$, for every k, $h(k)$ only depend on the last 2 bits.

  $k = 7 = 0111 \mod 0100 = 0011$ → Only the last two bits matter

The multiplication method:

$$h(k) = \lfloor m(kA \mod 1) \rfloor$$

k → input key, m → number of slot in hash tables, A → a constant, $0 < A < 1$

Advantage: m is not critical

Universal hashing:

Disadvantages for above: Fixed hash functions → Adversary put keys in the same slot

Main idea: Randomly choose hash functions

Method to solve hash collisions:

Open addressing:

Main idea: Do not use link list, but occupy the hash table itself

Probe: To find an empty slot

Note: Pay attention to delete a key when probing: Nil → Deleted

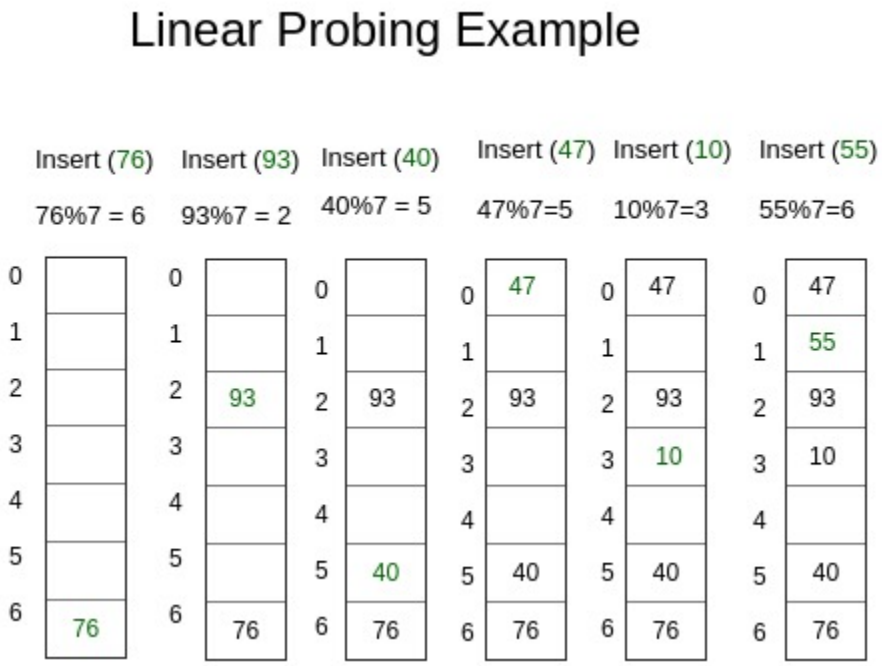
Linear probing:

$$h(k,i) = (h'(k) + i) \mod m$$

Pros: Simple to implement

Cons: Primary clustering → Occupied slots prolong the searching time

Example: If we have inserted 9 elements, which the result of hash function are all 0, then for the tenth element, you need to search 10 times until you find the empty slot.



Quadratic probing

$$h(k,i) = (h'(k) + c_1 i + c_2 i^2) \mod m$$

Difference: Just change the form of the step

Pros and Cons: Milder form of clustering → Secondary clustering

Double hashing:

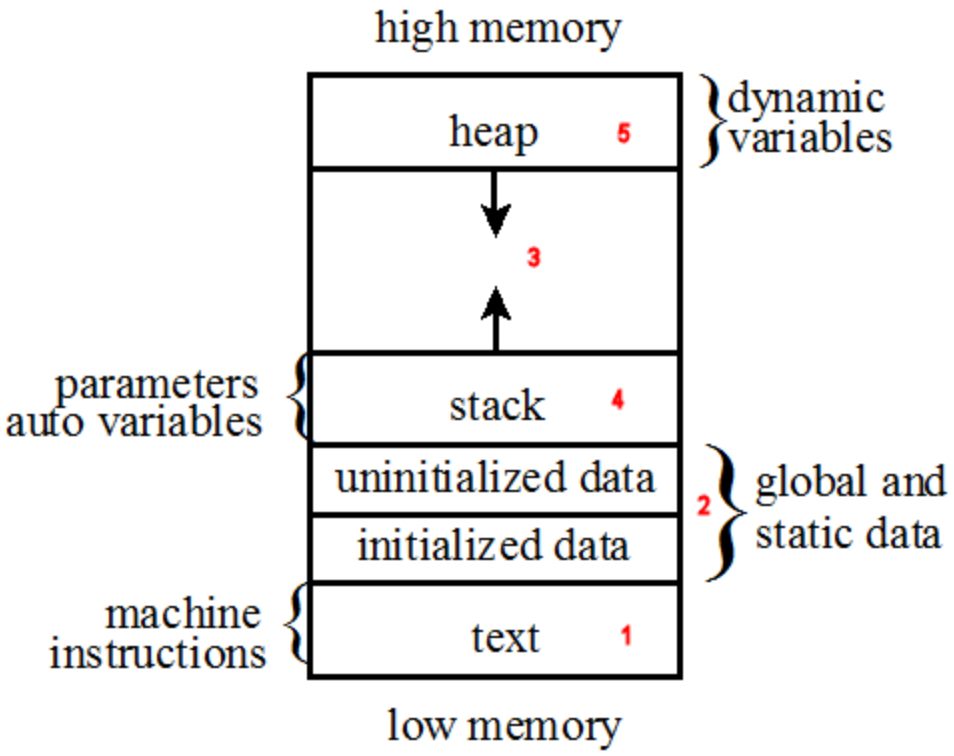
$$h(k,i) = (h_1(k) + i h_2(k)) \mod m$$

Differences: Both Probe sequence & initial position may vary

Stack and Heap

[Optional] Just give you some basic idea of stack and heap and help you understand what the professor said in the prevous lecture. You'll get more info in the OS course afterwards.

Section of memory → Reflect how the OS manage the process



Differences between stack and heap:

- Contents are different
- **Stack** is allocated and freed by the OS, **Heap** is controled by programmers → Easily leads to memory leak with careless usage
- Direction of the growth is different
- Heap is allocated dynamically, Stack can be allocated both dynamically or statically, but controled by OS
- Stack is supported by hardware(specific registers to store the address of the stack), heap is managed by functions in library of C/C++ → Stack is more efficient but heap is more flexible

FAQ for hw2

```
// 1.
segementation fault: 11
// It mainly because you access the memory in a wrong way
// Try to figure out whether you have already covered the corner cases especially
// when pointer points to NULL or nullptr
```

```
Node *ptr = nullptr;
ptr->next = a; // Makes no sense

// 2.
// Do not use quotation mark in the path
./main stack "../input/stack/0.txt" test //Remove the double quotation mark

// 3.
// How to specifically find where my code goes wrong?
// 1 - [Optional] Run you program step by step or line by line -> Debugger
// 2 - [Optional] Use log information -> You can google it if you interest
// 3 - If you feel complicated above, just print something out in each function

// 4.
// How to read and write file
// Use stringstream and fstream in C++ lib
// You can find more information in discussion 3
```