

Hash Tables

Big O:

- space $O(n)$
- time
 - search worst $O(n)$, average $O(1)$
 - insert worst $O(n)$, average $O(1)$
 - delete worst $O(n)$, average $O(1)$

Advantages:

- faster than other structures on large entries
- efficient when max entries known (dont have to resize)

Disadvantages:

- have to resize for more data

Uses:

- associative arrays (arrays index through arbitrary strings)
- database indexing
- caches
- sets (?)
- object rep (key is method or object, value is pointer to member or method)

Properties:

- keys have to be hash able (able to compute numeric value from it)
- entries in no particular order

Creating a Hash Table:

```
public HashMap<Integer, Student> buildMap(Student[] students) {  
    HashMap<Integer, Student> map = new HashMap<Integer, Student>();  
    for (Student s : students) map.put(s.getId(), s);  
    return map;  
}
```

Notes:

- Alternative to array to implement stack and queue
- Allows any length
- can be made more efficient with better fit hash function