CS61BL Tutoring Session

Hashing & Heaps

Hongli (Bob) Zhao

Hash Table

- Idea:
    - *Amortized* constant time operations: `add, find, insert`
    - Achieved by using an appropriate hash function to evenly map elements into buckets
- Implementation:
    - External chaining: Given N elements and M buckets, make sure length of chain does not exceed N/M on average
    - Resize if average length gets too high, and rehash all elements using a bigger table
    - Demo: Worksheet Q2

Hash Functions

- A mapping from an object to an ***integer value***
- A hash function must be valid:
  - Determinism: two objects with the same contents must be hashed to the same value
  - Consistency: One object at two different times must be hashed to the same value
- A bad hash function doesn’t mean it’s invalid (e.g. hashing to the same value) → check the two properties
- A generally good hash function avoids collisions

Amortized Runtime

- When spread out evenly across all operations, runs in constant time; constant time on average
- Good for quick insertion and lookup; not good at range query (i.e. specifying an order)
- Runtime comparisons:
  - See Worksheet Q3
  - Good to memorize!

Priority Queue

- Idea: order sequential data with some importance measure
  - Useful for sorting → keep outputting the value with the highest priority
  - High priority value != High priority
- Operations:
  - insert(item, priorityValue), peek(), poll()
- How to implement a PQ?

Heap

- Idea: one way for implementing a PQ with log(N) operations.
    - A binary tree data structure (not a binary search tree)
    - Can do with an Array, or Linked Lists as long as heap properties are satisfied (for Array: i -> 2i, 2i+1)
- Properties / Invariants:
    - Needs to be a complete tree (always bushy)
    - Needs to maintain the *heap property*:
        - Labels of both children are larger/less than the root (looser requirement than BST)
- Operations:
    - insert, removeMin/removeMax, findMin/findMax
    - reHeapifyUp and reHeapifyDown
    - Demo: Worksheet Q1
    - More Demo