UCLA Samueli Computer Science

CS32: Introduction to Computer Science

Discussion Week 10

Junheng Hao, Ramya Satish

Mar 15, 2019

Announcement

UCLA Samueli Computer Science

- **Final exam: Tomorrow!**
  - **From 11:30 am to 2:00 pm** in **Dodd 147** if you're enrolled in lecture 1 (Nachenberg).
  - **From 11:30 am to 2:00 pm** in **La Kretz 110** if you're enrolled in lecture 2 or 3 (Smallberg) and your last name begins with F, H, J, K, L, P, R, T, V, W, X, or Y.
  - **From 11:45 am to 2:15 pm** in **WG Young CS50** if you're enrolled in lecture 2 or 3 (Smallberg) and your last name begins with D, U, M, B, Z, O, M, B, I, E, G, A, S, C, A, N, or Q.
  - The exam is open book, open notes, no electronic devices. If you're the kind of person who asks questions during an exam, please sit in the front row or in an aisle seat. Bring a No. 2 pencil to the exam.

Outline

UCLA Samueli Computer Science

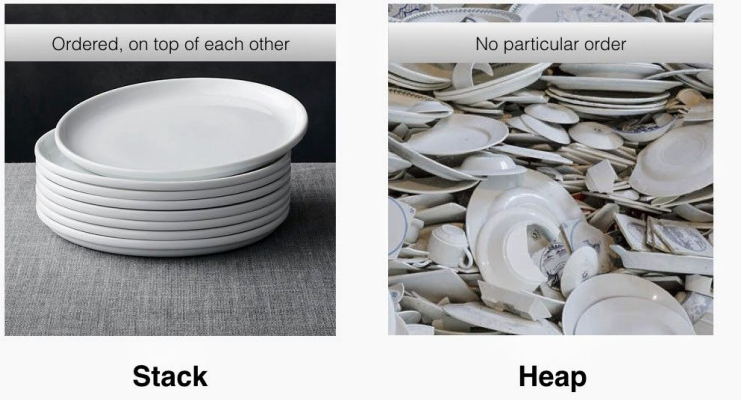
- Heap (Review), Priority Queue
- Graph
- Final exam review

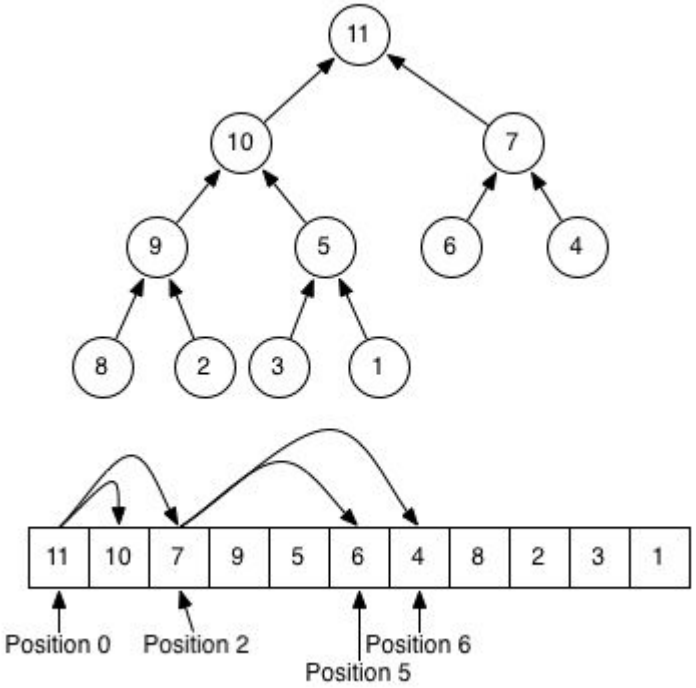
Heap

Definition and properties

UCLA Samueli Computer Science

- About heap
  - Heap is considered as complete binary tree.
  - Every nodes carries a value greater than or equal to its children (for MaxHeap).
  - Often implemented as an array.
  - Body structure of priority queue.

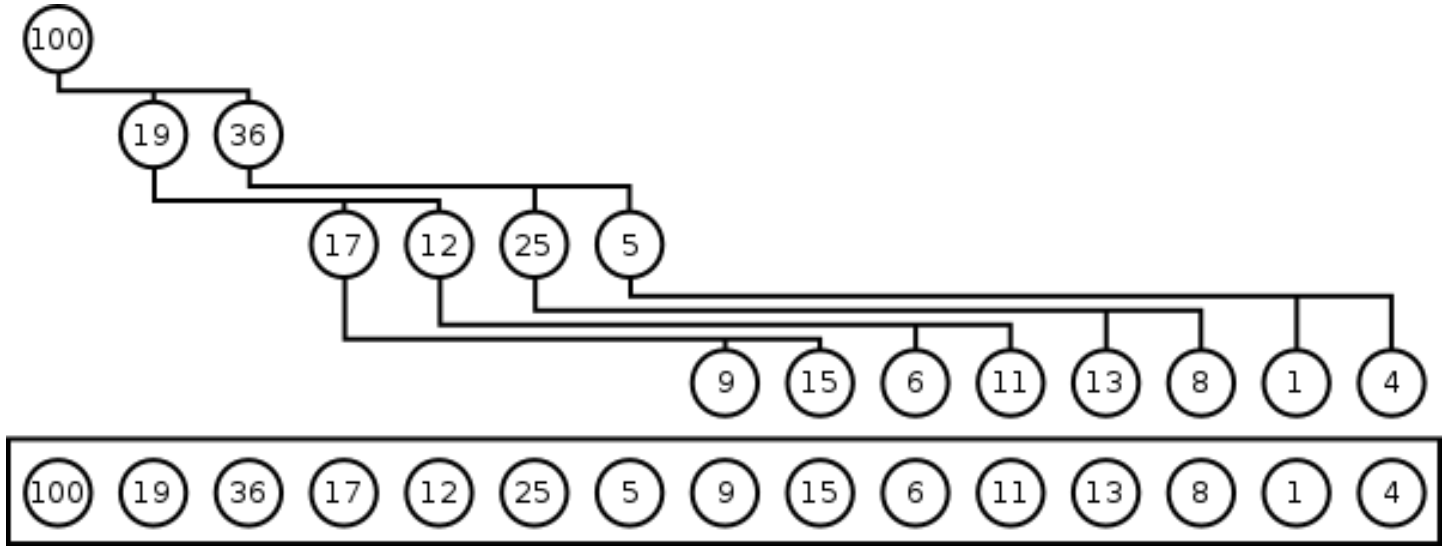




Heap

Implementation by arrays

UCLA Samueli Computer Science



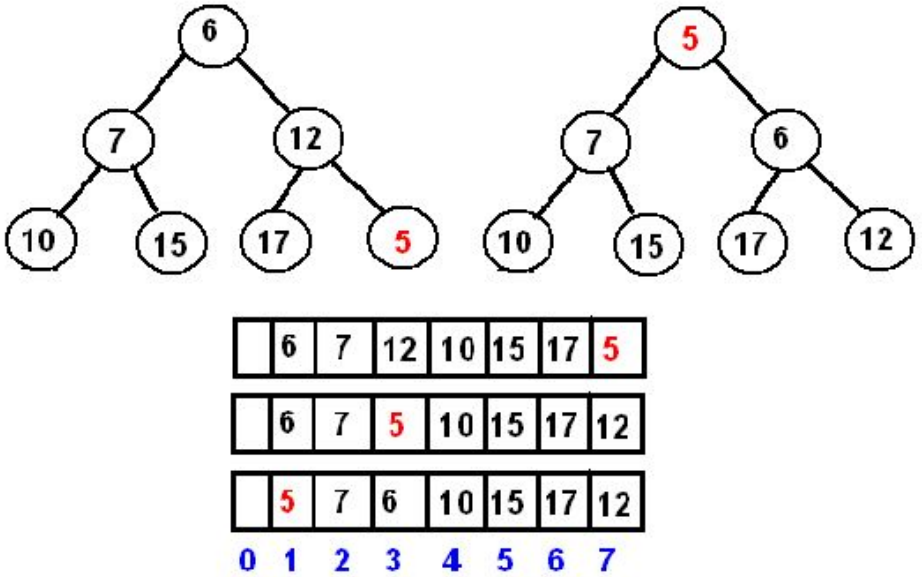
(Almost) full binary tree in an array implementation.

Heap

Insertion (MinHeap)

UCLA Samueli Computer Science

```
void insert(Comparable x)
{
  if(size == heap.length - 1) doubleSize();
  //Insert a new item to the end of the array
  int pos = ++size;
  //Percolate up
  for(; pos > 1 && x.compareTo(heap[pos/2]) < 0;
pos = pos/2 )
    heap[pos] = heap[pos/2];
  //Finish
  heap[pos] = x;
}
```



(Average/Worst) Complexity: $O(\log n)$

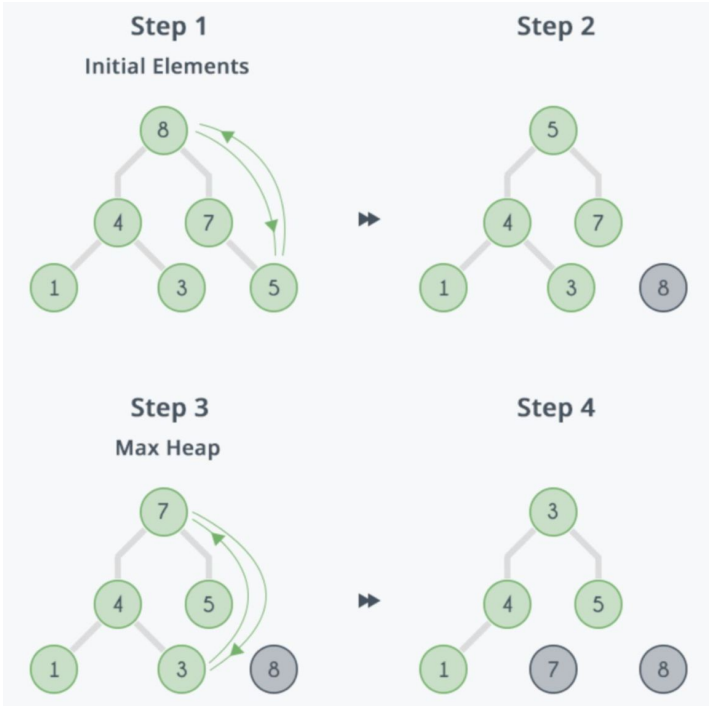
Credit to: https://www.cs.cmu.edu/~adamchik/15-121/lectures/Binary%20Heaps/heaps.html

Heap

Delete Min in MinHeap & HeapSort

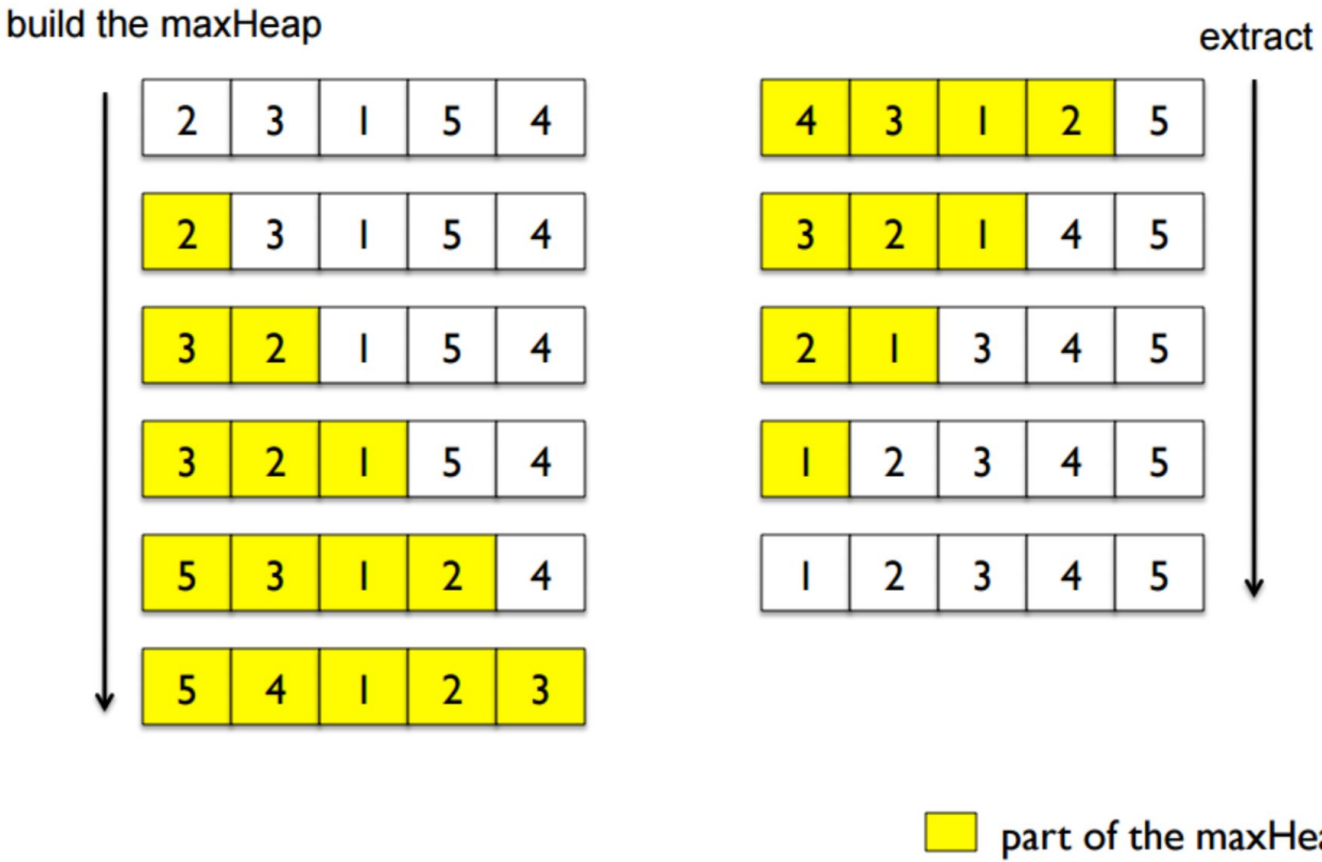
UCLA Samueli Computer Science

- The minimum element can be found at the root, which is the first element of the array.
- We remove the root and replace it with the last element of the heap and then restore the heap property by percolating down.
- (Average/Worst) Complexity: $O(\log n)$, same as insertion.



In-place Heapsort (with an array)

UCLA Samueli Computer Science



(Average/Worst) Complexity:

$$O(n \log n)$$

6 5 3 1 8 7 2 4

Heap

MinHeap and MaxHeap

UCLA Samueli Computer Science

- How can we efficiently find ***k* largest numbers** from n numbers? (n>>k, but k is not small)
  - Sort? → O(*n*log*n*)
  - Scan *k* times by linear search? → O(*nk*)
- Use heapsort
  - Only keep the largest
  - Whether to use MaxHeap or MinHeap?
  - Overall complexity?

Min Heap!
Pop out the `min` from heap and insert a number, if it's larger than `min`.

O(*n*log*k*)

Heapsort Exercise

Find median from a streaming data

UCLA Samueli Computer Science

How to find the median of the streaming data?

That is, implementing the following program:

```
addNum(1)
addNum(2)
findMedian() -> 1.5
addNum(3)
findMedian() -> 2
```

Here you may use two heaps to store all the coming data → find median in O(1) time.

```
class MedianFinder {
public:
  MedianFinder() {
    // construction
  }

  void addNum(int num) {
    // add new integers from stream
  }

  double findMedian() {
    // return the median
  }
private:
    // define your private data member(s)
};
```

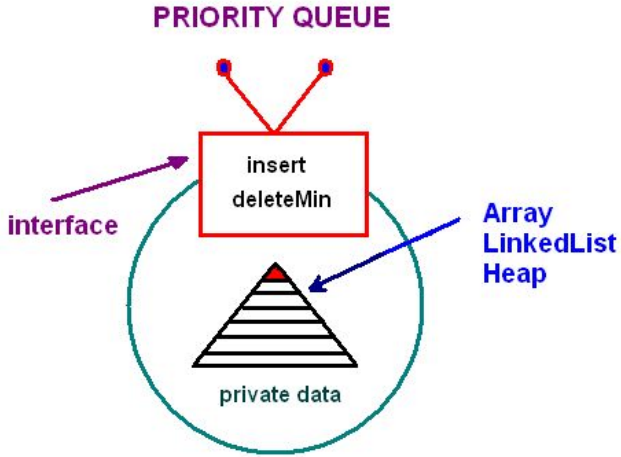
Priority Queue

Defininations & Properties

UCLA Samueli Computer Science

- Abstract data type (providing interface)
- Heap-based Priority Queue over other implementations:

Implementations	Insert	DeleteMin	FindMin
Ordered array	O(n)	O(1)	O(1)
Ordered list	O(n)	O(1)	O(1)
Unordered array	O(1)	O(n)	O(n)
Unordered list	O(1)	O(n)	O(n)
Binary Heap	**O(log n)**	**O(log n)**	**O(1)**



Credit to: https://www.cs.cmu.edu/~adamchik/15-121/lectures/Binary%20Heaps/heaps.html

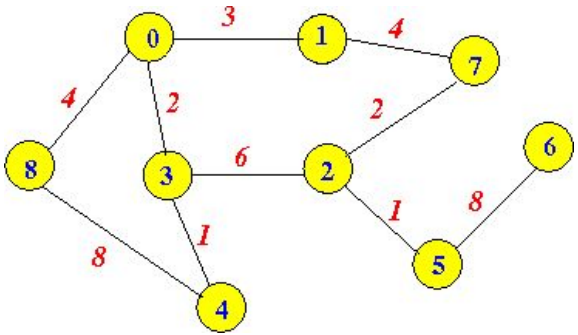
Graph

Introduction, BFS & DFS

UCLA Samueli Computer Science

- Terms
  - Nodes, edges
  - Adjacency matrix, adjacency list
- Complexity to store graph, add/remove vertex, add/remove edge
- Graph algorithms
  - BFS: Breadth First Search Graph traversal algorithm
  - DFS: Depth First Search Graph traversal algorithm [Link]
  - Dijkstra's Algorithm: Compute the minimum cost paths from a node (e.g., node 1) to all other node in the graph [Link]
  - Prim's Algorithm: finding Minimum cost Spanning Tree [Link]
  - More in CS180

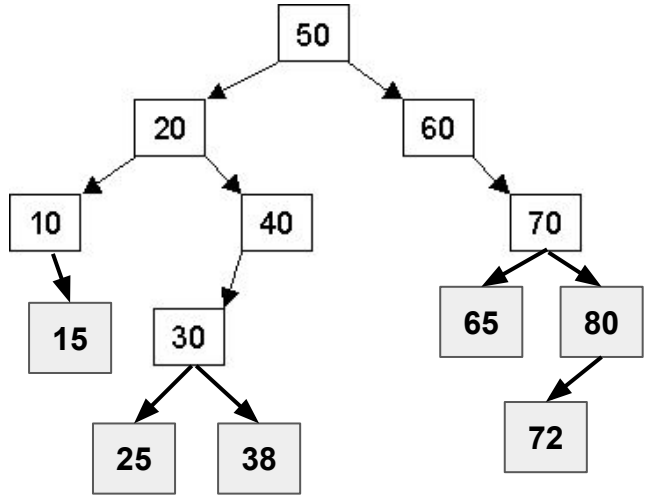
Any other methods to store a graph?
Incidence Matrix, etc.

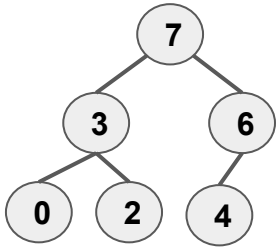


Homework 5 Review [Solution Link]

UCLA Samueli Computer Science

- Problem 1: BST
  - Insert, traversal, deletion
- Problem 2: Insert with parent node pointer
- Problem 3: Heap insertion and deleteMax
- Problem 4: Complexity Discussion





7	3	6	0	2	4	/

6	3	4	0	2	/	/

Final Review

UCLA Samueli Computer Science

- https://kycode.me/CS-32/
- http://getacollegelife.tumblr.com/post/70756494466/my-buddy-rachel-fangs-cs31-and-cs32-cheat
- Final exam practice
  - http://web.cs.ucla.edu/classes/winter19/cs32/Sampleproblems/ChangFinalPractice.pdf
  - http://web.cs.ucla.edu/classes/winter19/cs32/Sampleproblems/ChoiFinalPractice.pdf

Credit: Angelina Poole's personal page. https://sites.google.com/g.ucla.edu/angelinapoole

Final Review: Topics in CS32

You really have learned a lot!

UCLA Samueli Computer Science

- Modern features about C++
  - Resource management
  - Inheritance and polymorphism
  - Templates, Iterators, STL containers
- Data Structures
  - Array, Vector
  - Linked List
  - Stack, Queue
  - Tree, Heap, Graphs
  - Hash Table
- Algorithms and complexity analysis
  - Recursion
  - Big-O
  - Sorting

David's Reminders: Part 1

UCLA Samueli Computer Science

- Classes containing members of a class type -- order of construction and destruction, initializer lists.
- Destructors, copy constructors, and assignment operators.
- Base/derived classes and inheritance of members
- Automatic conversion of Derived* to Base* and Derived& to Base&
- Virtual functions, overriding member function implementations, pure virtual functions and abstract base classes, virtual
- Construction order and destruction order
- Recursion

David's Reminders: Part 2

UCLA Samueli Computer Science

- Various sorting algorithms.
- Preorder traversal and postorder, traversal and then introduced binary trees and more tree algorithms especially using recursion.
- Data structures and corresponding complexity (those Big-Os)
- Clear on open hash tables, load factor, and so on.
    - They should almost *never* assume that collisions are impossible. Even if you have a hash table with 10000 buckets and are storing only 100 items, you may well have 2 of those keys that happen to end up in the same bucket. It's a common beginner mistake to write code that assumes that if a bucket is not empty, you've found the item and the first item in the bucket is it.

David's Reminders: Not in CS32 final

UCLA Samueli Computer Science

- Multiple inheritance, private inheritance, protected members in inheritance
- Details of AVL trees, 2-3 tree or red-black trees, or more advanced trees.
- Specific hash function (FNV-1)
- Graphs
- Accurate names of member function of STL containers.

"The important thing to know is imply that there exist algorithms that keep the trees more or less balanced so that the average and worst case insert, delete and lookup performance is **O(log *n*)**." -- David

Helpful resources

UCLA Samueli Computer Science

- TA Links
  - Youfu Li [Link], Angelina Poole [Link], Jack Gong [Link], Trevor Hackett [Link], Qianru Li [Link], Arghya Mukherjee [Link]
  - Jin Wang, Ling Ding, Aalisha Dalal on CCLE
- Previous TA Mark Edmonds's CS32 worksheet (Spring 2016): [Link]
- UCLA CS Practice Problems [Link]
- Final exam practice from instructors. Check announcements here! [Link]
- Your virtual CS31 TA. [Link] *(Note: Sometimes it is helpful!)*

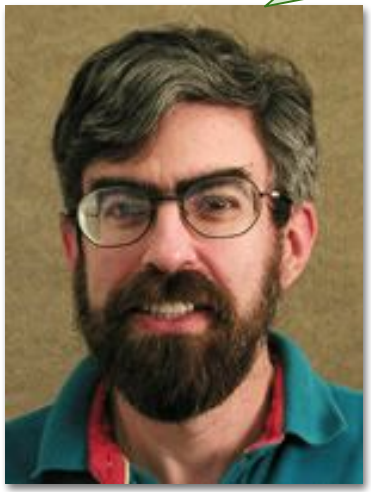
Credit: Angelina Poole's personal page. https://sites.google.com/g.ucla.edu/angelinapoole

UCLA Samueli Computer Science

Thank you!

Q & A

KEEP
CALM
AND
GOOD
LUCK



Group Exercise

UCLA Samueli Computer Science

There is no new worksheet this week.

Please check worksheet 6 and two final practice exams.