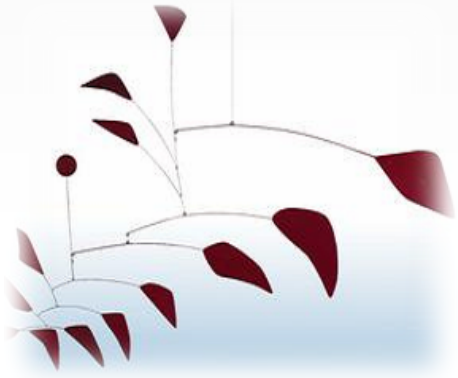


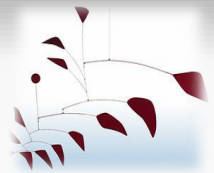
EECS 4101-5101
Advanced Data Structures

Shahin Kamali

A Quick Review and Last Words

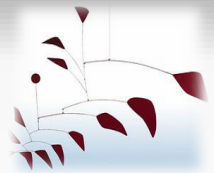
York University

Picture is from the cover of the textbook CLRS.



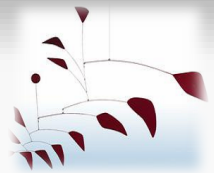
Summary of Topics

- **Introduction:**
  - **Amortization:** consider complexity in the long-run; don't focus on a single operation.



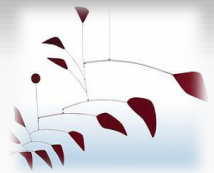
Summary of Topics

- **Introduction:**
  - **Amortization:** consider complexity in the long-run; don't focus on a single operation.
  - **Self-Adjustment:** update your data structure to reflect on patters seen in the input.
  - **Competitiveness:** most decisions in the real-world are online; try to settle worst-case guarantees.



Summary of Topics

- **Dictionaries:**
  - Move-to-Front Heuristic on Linear Lists.



Summary of Topics

- **Dictionaries:**
  - Move-to-Front Heuristic on Linear Lists.
  - Binary Search Trees, AVL Trees, Red Black Trees.

Summary of Topics

- **Dictionaries:**
  - Move-to-Front Heuristic on Linear Lists.
  - Binary Search Trees, AVL Trees, Red Black Trees.
  - B-trees, 2-3-4 Trees.

Summary of Topics

- **Dictionaries:**
  - Move-to-Front Heuristic on Linear Lists.
  - Binary Search Trees, AVL Trees, Red Black Trees.
  - B-trees, 2-3-4 Trees.
  - Splay Trees, Dynamic Optimality Conjecture.

Summary of Topics

- **Dictionaries:**
  - Move-to-Front Heuristic on Linear Lists.
  - Binary Search Trees, AVL Trees, Red Black Trees.
  - B-trees, 2-3-4 Trees.
  - Splay Trees, Dynamic Optimality Conjecture.
  - Hash tables.

Summary of Topics

- **Dictionaries:**
  - Move-to-Front Heuristic on Linear Lists.
  - Binary Search Trees, AVL Trees, Red Black Trees.
  - B-trees, 2-3-4 Trees.
  - Splay Trees, Dynamic Optimality Conjecture.
  - Hash tables.
  - Augmenting Data Structures.

Summary of Topics

- **Multi-Dimensional Dictionaries:**
  - Quad Trees: good for uniformly distributed data; bad in the worst-case.

Summary of Topics

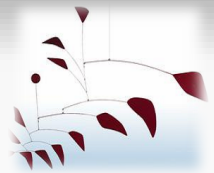
- **Multi-Dimensional Dictionaries:**
  - Quad Trees: good for uniformly distributed data; bad in the worst-case.
  - k-d trees: range search query running time is $O(\sqrt{n} + k)$ in 2-dimension.

Summary of Topics

- **Multi-Dimensional Dictionaries:**
  - Quad Trees: good for uniformly distributed data; bad in the worst-case.
  - k-d trees: range search query running time is $O(\sqrt{n} + k)$ in 2-dimension.
  - Range trees: the running time is improved to $O(\log^2 n + k)$.

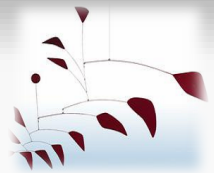
Summary of Topics

- **Multi-Dimensional Dictionaries:**
  - Quad Trees: good for uniformly distributed data; bad in the worst-case.
  - k-d trees: range search query running time is $O(\sqrt{n} + k)$ in 2-dimension.
  - Range trees: the running time is improved to $O(\log^2 n + k)$.



Summary of Topics

- **Priority Queues:** applications in algorithms such as Dijkstra's shortest path.
  - Binary Heaps: support insert, extract-min (or extract-max), and decrease-key in $O(\log n)$.

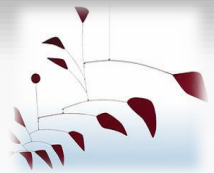


Summary of Topics

- **Priority Queues:** applications in algorithms such as Dijkstra's shortest path.
  - Binary Heaps: support insert, extract-min (or extract-max), and decrease-key in $O(\log n)$. Not good for merge.
  - Binomial Heaps: As good as binary heaps, support merge in $O(\log n)$ too.

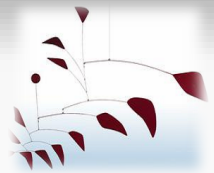
Summary of Topics

- **Priority Queues:** applications in algorithms such as Dijkstra's shortest path.
  - Binary Heaps: support insert, extract-min (or extract-max), and decrease-key in $O(\log n)$. Not good for merge.
  - Binomial Heaps: As good as binary heaps, support merge in $O(\log n)$ too.
  - Fibonacci Heaps: improve the amortized running time to a constant.



Summary of Topics

- **Disjoint Sets:**
  - Disjoint sets have applications in algorithm design, e.g., in Kruskal's MST algorithm.



Summary of Topics

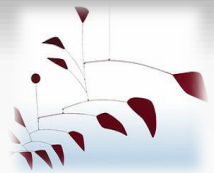
- **Disjoint Sets:**
  - Disjoint sets have applications in algorithm design, e.g., in Kruskal's MST algorithm.
  - union and find operations can be supported in **almost** constant amortized time, using path-compression and union-by-rank.

Summary of Topics

- **Randomized Data Structures:**
  - Skip lists and Treaps offer simple randomized data structures for dictionaries with similar guarantees as Red-Black and AVL trees.

Summary of Topics

- **Randomized Data Structures:**
  - Skip lists and Treaps offer simple randomized data structures for dictionaries with similar guarantees as Red-Black and AVL trees.
  - Randomized data structures offer security against adversarial attacks, and are often can be reflective to predictions about the input queries.



Summary of Topics

- **String Data Structures:**
  - Use Patricia trees to maintain dictionaries of strings.

Summary of Topics

- **String Data Structures:**
  - Use Patricia trees to maintain dictionaries of strings.
  - Store a single string in a suffix tree to answer queries such as pattern matching.

The Ending

Observation

You should aim for the stars - and hopefully avoid ending up in the clouds! Roxanne McKee

- Template for final will be posted. If any thing in the slides is not clear, ask me to explain it on Piazza.

The Ending

Observation

You should aim for the stars - and hopefully avoid ending up in the clouds! Roxanne McKee

- Template for final will be posted. If any thing in the slides is not clear, ask me to explain it on Piazza.
- Your feedback is appreciated; if something can be improved (which is 100 percent the case), let me know.

The Ending

Observation

You should aim for the stars - and hopefully avoid ending up in the clouds! Roxanne McKee

- Template for final will be posted. If any thing in the slides is not clear, ask me to explain it on Piazza.
- Your feedback is appreciated; if something can be improved (which is 100 percent the case), let me know.
- I hope to see you in future courses.