

Course Review

Note Title

12/11/2007

Graphs

Graph Representations

Adjacency List

Adjacency Matrix

Given a small graph, you should be able to illustrate these

Choose which is appropriate for an application

Graph Algs

Single Source Shortest Path Algs

Common use:

vertex \leftrightarrow state in world

edge \leftrightarrow action that moves
you between states

directed vs undirected
weighted vs unweighted

See when you can model a problem as a shortest path problem

Unweighted - BFS starting

$O(n+m)$ at source (if there's a goal you can stop when you discover a path)

weighted -

Fibonacci Dijkstra's alg - No negative weight edges are allowed
 $O(n \log n + m)$

Minimum Spanning Tree

Prim's + Kruskal's Alg

Depth First Search

topological sort

in-place dfs + use in

garbage collection (Mark + Sweep)

Trie-based data structure

find all elements with common prefix

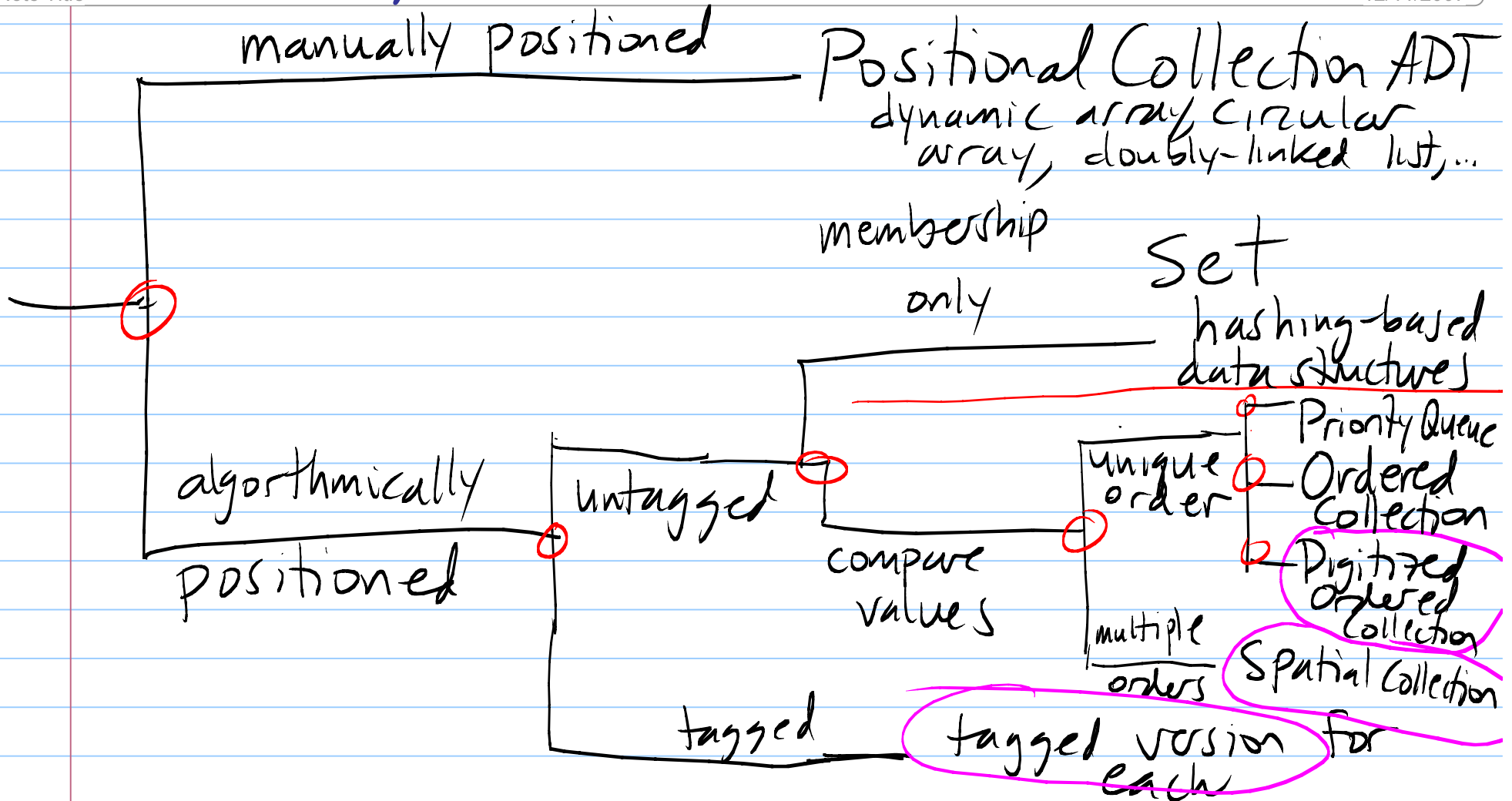
$$O\left(\# \text{ digits in prefix} + \# \text{ elements returned} \times b\right)$$

alphabet
size
(base)

Taxonomy of ADTs (Secs 2.1-2.6)

Note Title

12/11/2007



Spatial Collections

Kd-tree + quad tree

$$O\left(n^{\frac{k-1}{k}} + \frac{\# \text{ elements}}{n \text{ box}}\right)$$

↑
all elements
in box

Digitized Ordered Collection

trie, compactTrie, compressedTrie

↓ at high-level understand

adv + disadv to Ordered Collection
data structures

Ordered Collection

Binary Search Tree

Balanced Search Tree (Rotation)

Red-Black Tree properties & insertion
I won't ask about deletion

B-trees (+ brief intro to B⁺-tree)

I won't ask about deletion (except in really easy case like in homework)

↑
only need to merge (reverse split)

Priority Queue
Binary Heap

Focus is up to here
3/4

Covered on Mid term

Hashing-based data structures
Open Addressing + Separate chaining

+ Also Direct Addressing

Sorting Algs

radix sort + counting sort

quick sort

merge sort

insertion

Know how these work, pros + cons,
+ which is the best choice

Adversary Lower Bound

Designing a strategy for an adversary to answer questions

Chosen by an algorithm

to show how many questions any alg must ask so adv. only

has one input left consistent with all past answers.

Asymptotic notation

Divide & conquer algs
master method