CS 171: Introduction to Computer Science II

Project Workshop and Review

Li Xiong

Today

- Project workshop
- Final review
- Course evaluation

Project Workshop

- FaceSpace
  - Alex Fields and Andres Celis
- MapQuest
  - Kenty Wang
  - Jennifer Lin
  - Xiaobo Sun

Project Workshop

- Project design – class design, algorithm design, data structures, GUI design
- Project development – development, debugging, integration
- Project experience – interesting and challenging aspects, lessons learned
- Application Demo – program features, desired features/remaining work

Review

- Data structures
- Algorithms
- When to use what

Summary of Data Structures

- General purpose data structures
  - Arrays
  - Linked lists
  - Trees
  - Hash tables
- Specialized data structures
  - Stacks
  - Queues, priority queues
  - Graphs

Summary of Algorithms

- Associated with each data structure
  - Insert
  - Delete
  - Search
  - Traversal
  - Graph algorithms
- Sorting
  - Bubble, insertion, selection
  - Merge sort
  - Quick sort

Algorithm Analysis

- Big-O notation
- Common functions
  - constant, logarithm, linear, quadratic, polynomial, exponential, factorial in increasing order of growth
- Cost analysis
  - Direct methods
  - Recursive relations (specially useful for recursive algorithms)

Programming/Problem Solving Techniques

- Recursion
- Divide and conquer
- Backtracking
- Dynamic programming – memoization

When to Use What?

- General-purpose data structures
  - Arrays (unordered)
  - Ordered arrays

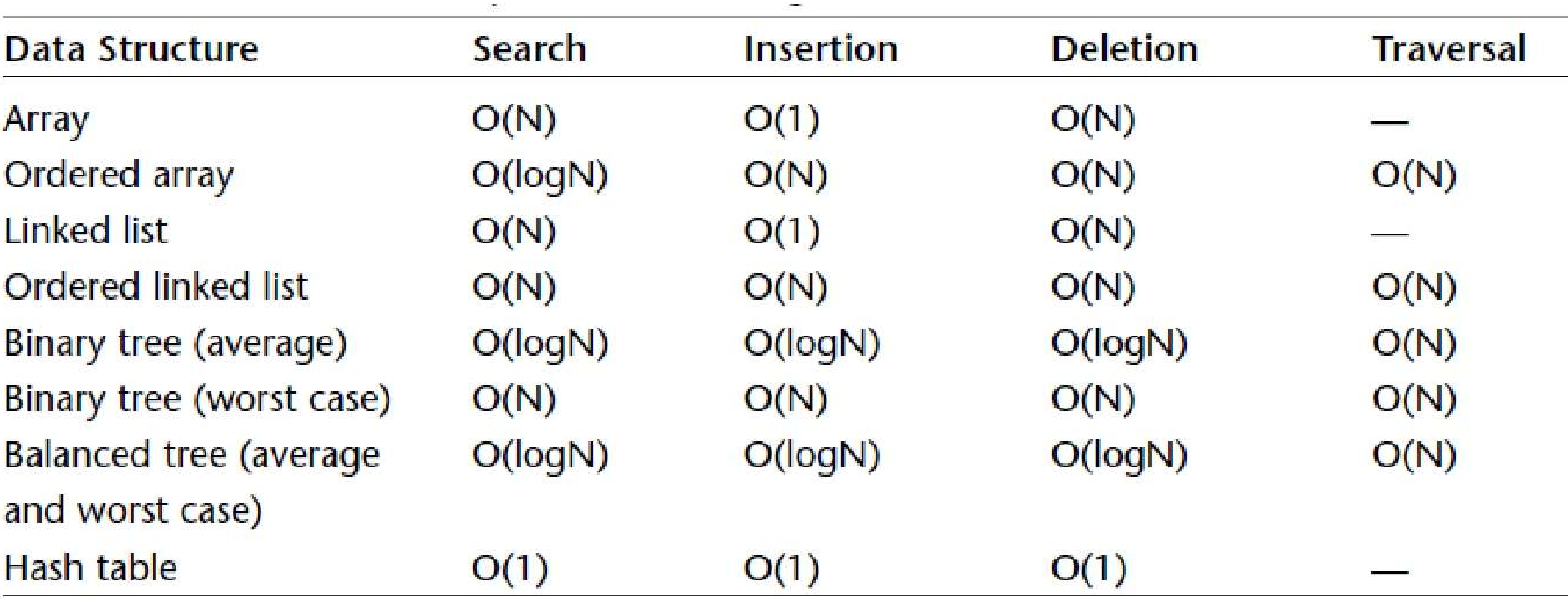
  - Linked list (unordered)
  - Ordered Linked list

  - Binary search tree (unbalanced)
  - Self-balancing binary tree

  - Hash tables

When to Use What?

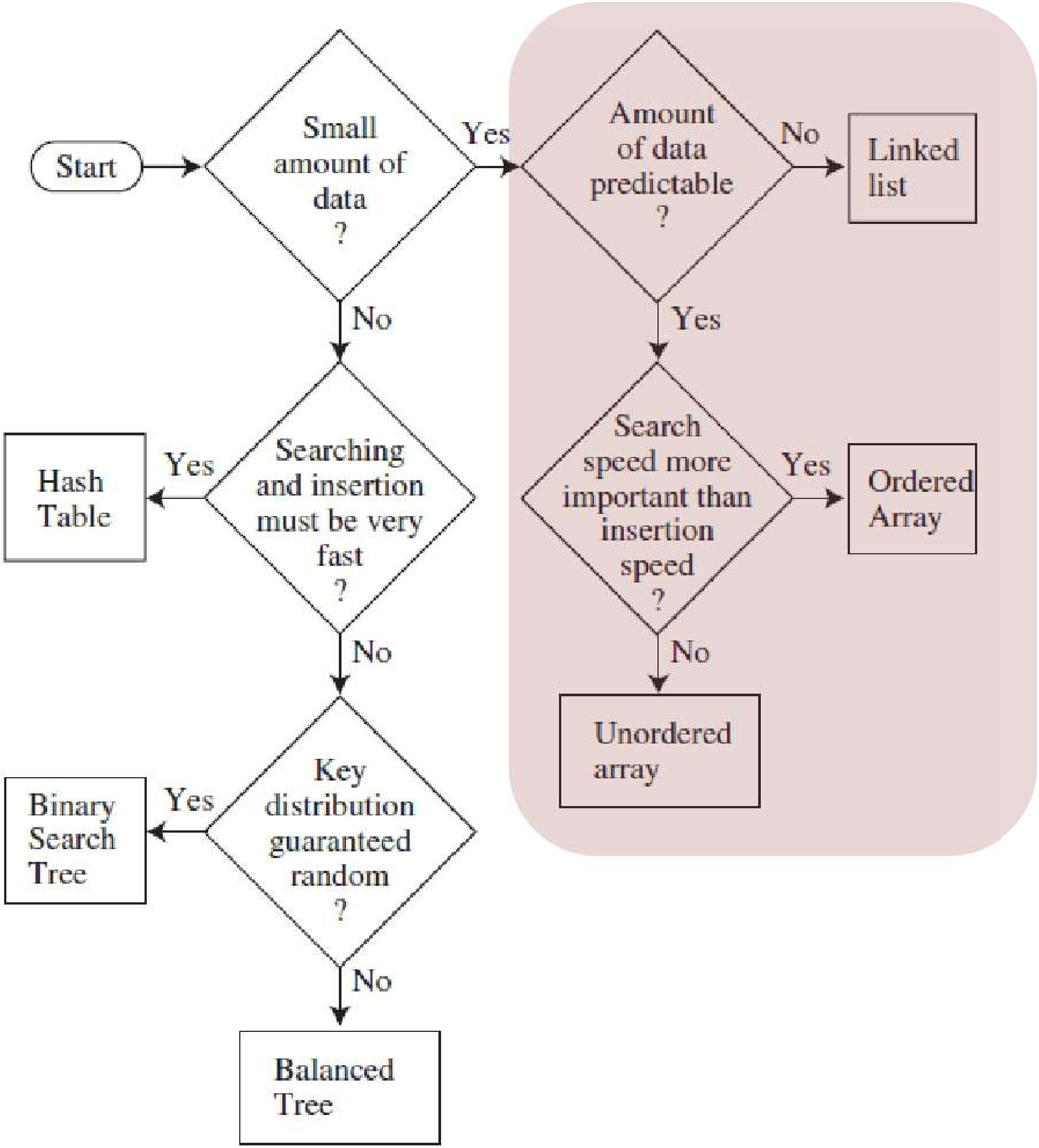
- **Comparison of General Purpose Structures:**

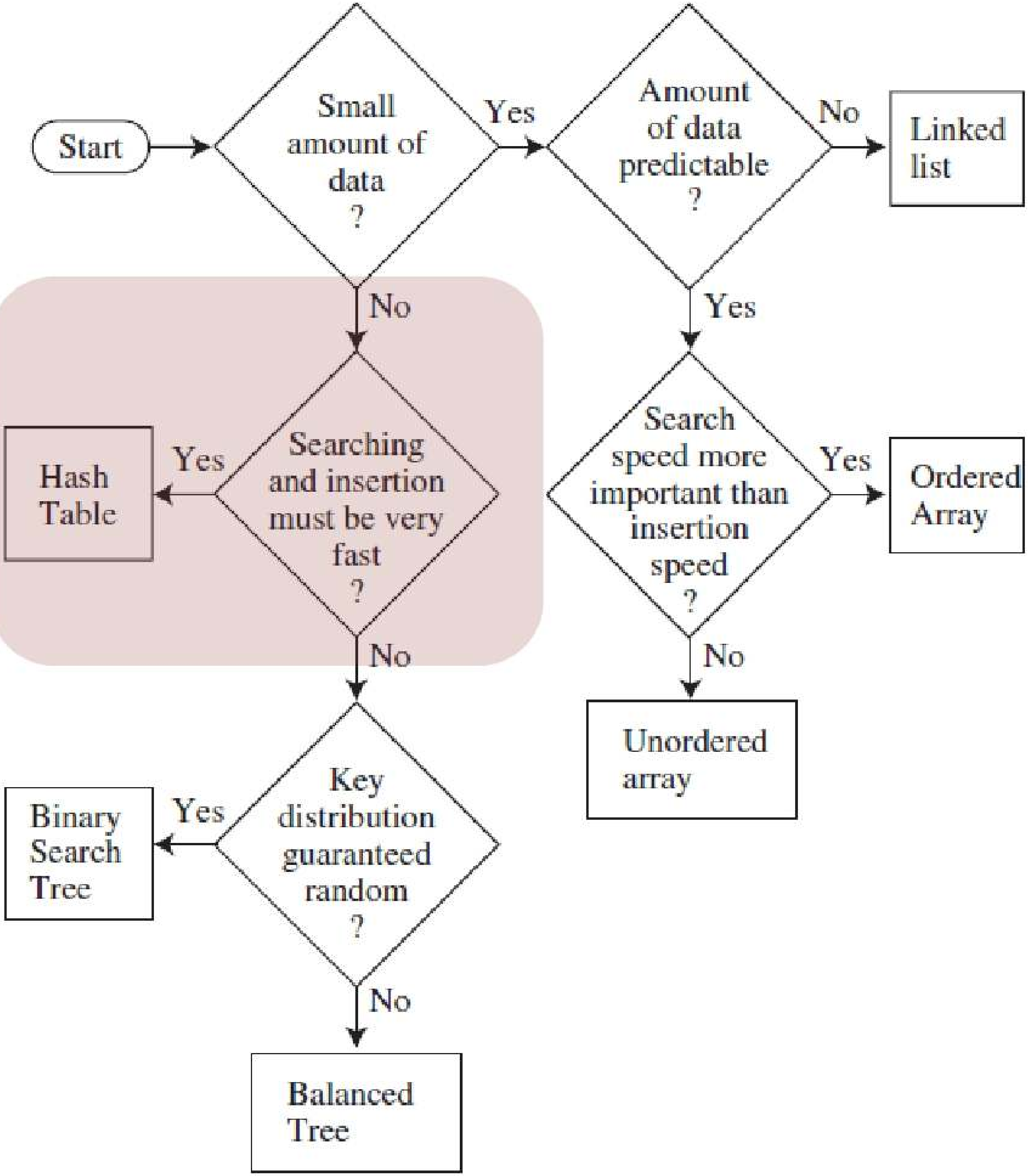


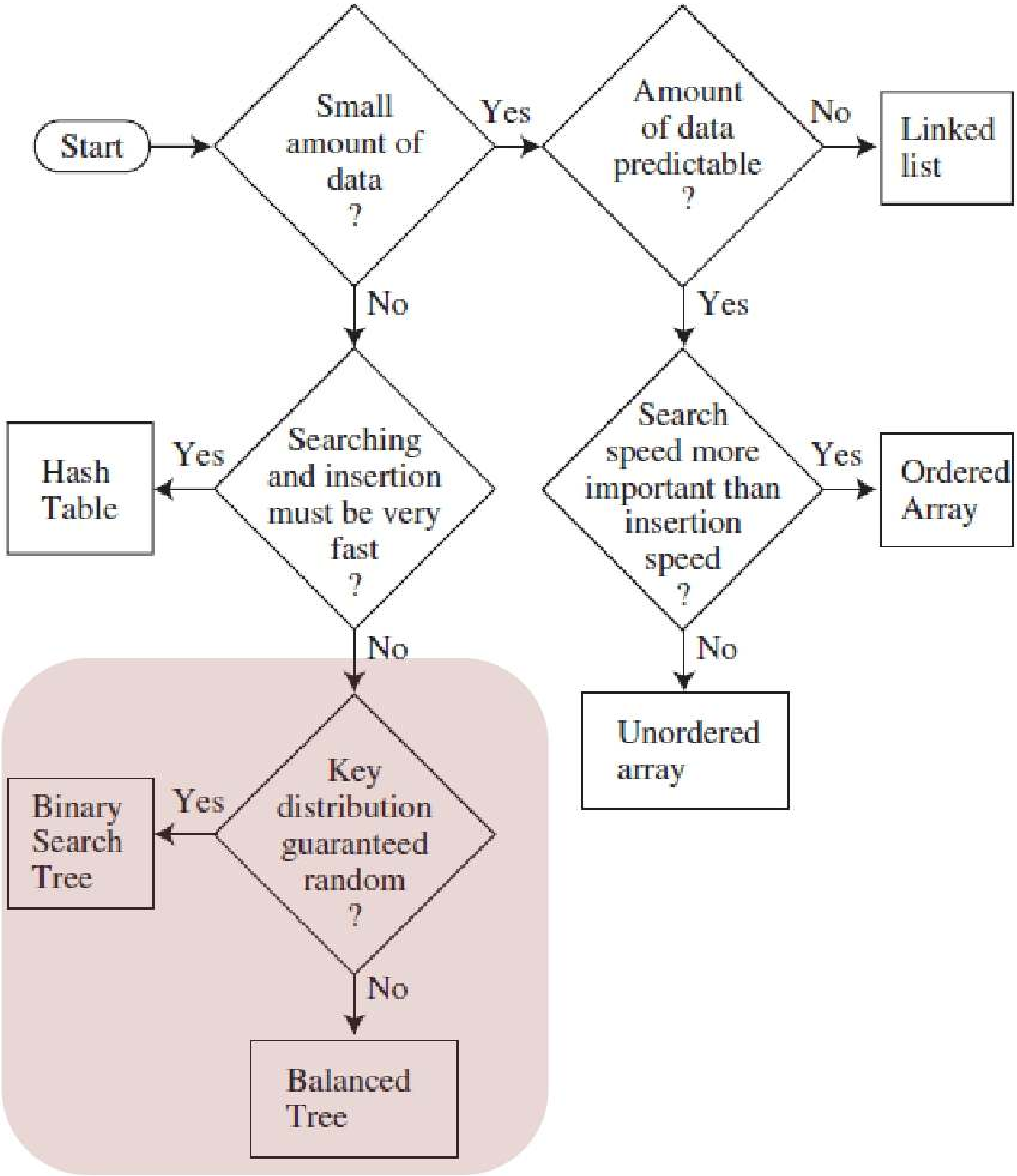
Data Structure	Search	Insertion	Deletion	Traversal
Array	O(N)	O(1)	O(N)	—
Ordered array	O(logN)	O(N)	O(N)	O(N)
Linked list	O(N)	O(1)	O(N)	—
Ordered linked list	O(N)	O(N)	O(N)	O(N)
Binary tree (average)	O(logN)	O(logN)	O(logN)	O(N)
Binary tree (worst case)	O(N)	O(N)	O(N)	O(N)
Balanced tree (average and worst case)	O(logN)	O(logN)	O(logN)	O(N)
Hash table	O(1)	O(1)	O(1)	—

When to Use What

- Speed
  - Array and Linked Lists < Trees < Hash Table for large amount of data
- Catch
  - Trees
    - Binary search tree may be imbalanced
    - Balanced trees are complex
  - Hash tables
    - A good hash function may be difficult to achieve
    - Hard to expand (dynamic resizing of the array)
    - Performance degrade when table is too full
    - Can’t store data in sorted order


Start
Small amount of data ?
Yes
Amount of data predictable ?
No
Linked list
Yes
Search speed more important than insertion speed ?
Yes
Ordered Array
No
Unordered array
No
Searching and insertion must be very fast ?
Yes
Hash Table
No
Key distribution guaranteed random ?
Yes
Binary Search Tree
No
Balanced Tree


Start
Small amount of data ?
Yes
Amount of data predictable ?
No
Linked list
No
Yes
Searching and insertion must be very fast ?
Yes
Hash Table
Search speed more important than insertion speed ?
Yes
Ordered Array
No
No
Unordered array
Key distribution guaranteed random ?
Yes
Binary Search Tree
No
Balanced Tree



Start
Small amount of data ?
Yes
No
Amount of data predictable ?
No
Linked list
Yes
Searching and insertion must be very fast ?
Yes
Hash Table
No
Search speed more important than insertion speed ?
Yes
Ordered Array
No
Unordered array
Key distribution guaranteed random ?
Yes
Binary Search Tree
No
Balanced Tree

Specialized data structures

- Stacks
  - Push, pop
- Queues
  - Insert, remove
- Priority queues
  - Insert, removeMin/removeMax
  - Binary heaps
- Graphs
  - List of edges, adjacency matrix, adjacency list
  - BFS and DFS traversal algorithms
  - Shortest paths for unweighted graphs (BFS)
  - Shortest path for weighted graphs (Dijkstra, A*)

Final Exam

- May 7, 4:30 – 7pm, W303
- Exam format and difficulty level are similar to mid-term, quizzes, practice exam
- It is accumulative: 1/3 before midterm, 2/3 after midterm
- Brief review guide and practice exam are posted

Parting thoughts

- You have learned a great deal!
- Beginning at the end
- Thank you for a great semester and would love to hear from you or see you in a future class!