

Choosing a good ADT

10/26/04 22:37

Choosing a good ADT

1

Table

- ◆ Store several pieces of information together
 - First and last name, telephone number, address, etc
- ◆ Each group is called a record
- ◆ You choose what ADT to store the records in
 - List or tree

10/26/04 22:37

Choosing a good ADT

2

Choosing a Good Key

- ◆ Choose one piece of information in the record to sort on - choose the key
- ◆ Think about what most of the lookups will be based on
- ◆ Think about operations to be done
 - Create, destroy, empty, insert, delete, search, print, etc

10/26/04 22:37

Choosing a good ADT

3

Choosing a Good Implementation

- ◆ Linear implementation
 - Sorted or unsorted
 - Array or pointer based
- ◆ Non-linear
 - Tree, binary tree, BST
 - Array or pointer based
- ◆ Choose based on the use for the ADT
 - What operations and how often

10/26/04 22:37

Choosing a good ADT

4

Scenario A

- ◆ Insert and print in no particular order
- ◆ Unsorted is acceptable
- ◆ Search, delete, and sorted traversal are rare
 - I.e. grocery list - record then print later

10/26/04 22:37

Choosing a good ADT

5

Scenario A

- ◆ Insertions - unsorted
 - Array - $O(1)$
 - Linked list - $O(1)$
- ◆ Do you choose array or linked list?
 - Do you know the expected size ahead of time?
- ◆ Is a BST a good idea?
 - Insertions - $O(\log n)$ on average

10/26/04 22:37

Choosing a good ADT

6

Scenario B

- ◆ Frequent searching
 - I.e. Thesaurus, dictionary, directory, etc
- ◆ Efficient search needed
 - Not many insertions or deletions expected
- ◆ Binary search on a linked list?
 - How do you find middle element?
 - ◆ Traverse $n/2$ items at each iteration
- ◆ Binary search on an array is more efficient - $O(\log n)$

10/26/04 22:37

Choosing a good ADT

7

Scenario B

- ◆ Do you know the size?
 - Use array if size is known, BST if unsure

10/26/04 22:37

Choosing a good ADT

8

Scenario C

◆ Insert, delete, search, and print sorted

- I.e. Car dealership
 - ◆ Receive and sell cars frequently
 - ◆ Search for a car
 - ◆ Print inventory
 - ◆ Sort my model? VIN? Price?

Scenario C

◆ Search

- Sorted Array - $O(\log n)$
- Linked List - $O(n)$
- BST - $O(\log n)$

◆ Insert

- Sorted Array - $O(n)$
- Linked List - $O(1)$
- BST - $O(\log n)$

Scenario C

◆ Delete

- Sorted Array - $O(n)$
- Linked List - $O(1)$
- BST - $O(\log n)$

◆ Print

- $O(n)$ for all

Scenario C

◆ Combining...

- Array - $O(\log n)$, $O(n)$, $O(n)$, $O(n)$
- Linked List - $O(n)$, $O(1)$, $O(1)$, $O(n)$
- BST - $O(\log n)$, $O(\log n)$, $O(\log n)$, $O(n)$

◆ Which do you choose?