

ECE 2574: Data Structures and Algorithms - Dictionary ADT

C. L. Wyatt

Today we will look at a very flexible ADT the Dictionary, and discuss potential implementations.

- ▶ Dictionary ADT
- ▶ AbstractDictionary
- ▶ Testing the Dictionary
- ▶ An implementation using SortedList
- ▶ An implementation using BinarySearchTree

Dictionary ADT

A dictionary maps keys to values. Also called a *map* or a *key-value store*.

Example: a natural language dictionary: keys are words, value are the entries (part of speech, definitions, etc).

It supports:

- ▶ isEmpty()
- ▶ getNumberOfItems()
- ▶ add(searchKey, newItem)
- ▶ remove(searchKey)
- ▶ clear()
- ▶ getItem(searchKey)
- ▶ contains(searchKey)
- ▶ traverse(visit)

Lets define an interface

See `abstract_dictionary.h`

Lets implement some tests

See `test_dictionary.h`

We can implement `AbstractDictionary` using the `SortedList`

See `sorted_array_dictionary.h`.

What is the complexity of the operations?

An improved implementation would use a BinarySearchTree

See `bst_dictionary.h`.

What is the complexity of the operations?

Next Actions and Reminders

- ▶ Read about Treaps

Enjoy your Thanksgiving Break.