LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY

Data Structure

Chapter 1: Introduction to Data Structure

What this course is about ?

Data structures: conceptual and concrete ways to organize data for efficient storage and efficient manipulation

Employment of this data structures in the design of efficient algorithms

Why do we need them ?

Computers take on more and more complex tasks

- Imagine: index of 8 billion pages ! (Google)

Software implementation and maintenance is difficult.

Clean conceptual framework allows for more efficient and more correct code

Why do we need them

Requirements for a good software:

- Clean Design
- Easy maintenance
- Reliable (no core dumps)
- Easy to use
- Fast algorithms

- Efficient data structures
- Efficient algorithms

Example

A collection of 3,000 texts with avg. of 20 lines each, with avg. 10 words / line

$\rightarrow$ 600,000 words

Find all occurrences of the word "*happy*"

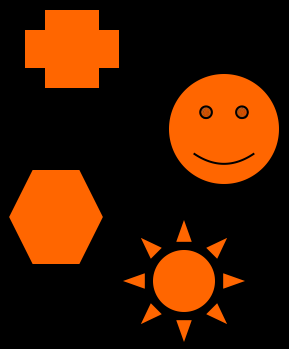
Suppose it takes 1 sec. to check a word for correct matching

What to do?

Example (cont'd)

Some example data structures

- $\log_2 600000 = 19$ sec. vs .166 hours!



What will you learn?

What are some of the common data structures

What are some ways to implement them

How to analyze their efficiency

How to use them to solve practical problems

What you need

- Programming experience with C / C++
  - Some Java experience may help as well (but not required)
- Textbook
  - Data Structures and Algorithm Analysis in C++
  - Mark Allen Weiss
- An Unix account to write, compile and run your C/C++ programs

Topics

Analysis Tools / ADT
Arrays
Stacks and Queues
Vectors, lists and sequences
Trees
Heaps / Priority Queues
Binary Search Trees –
Search Trees
Hashing / Dictionaries
Sorting
Graphs and graph algorithms