Data Structures & Algorithm Analysis

fall 2020

School of Software Engineering

South China University of Technology

Course Information

- Instructor
  - Yang XU
  - Email: xuyang@scut.edu.cn
- TA
  - Jun-ning TAN
  - Qian XIAO

- Schedule
  - Mondays 14:00 – 15:35 at A2-409
  - on Week 1- 12

Course Information (II)

- Lecture notes
  - Download here: http://eonline.jw.scut.edu.cn/meol/index.do

Course Information (III)

- Textbook
  - Mark Allen Weiss. Data Structures and Algorithm Analysis in C++ (4th Edition). Publishing House of Electronic Industry, 4th Edition

- Reference
  - 严尉敏 ,吴伟民. 数据结构 (C语言版) , 清华出版社
  - Sartaj Sahni. 数据结构、算法与应用 C++语言描述, 机械工业出版社

Course Information (III)

- Grading
  - Attendance and Assignments and programming projects 10%
  - Experiments 30%
  - Final exam 60%

Three Goals of the Course

- Master the commonly used data structures
  - They form a programmer’s basic data structure “toolkit”.
- Learn to measure the effectiveness of a data structure or algorithm.
  - Decide which data structure in the toolkit is most appropriate for a new problem
- Understand the idea of tradeoffs; reinforce the concept that costs and benefits are associated with every data structure.

Course Topics

- Overview
- Algorithm analysis
- Lists, Stacks, Queues
- Search Algorithms and Trees
- Hashing and Heaps
- Sorting
- Disjoint Sets
- Graph Algorithms

Data Structures: What?

- Generally, a data structure is any data representation and its associated operations.

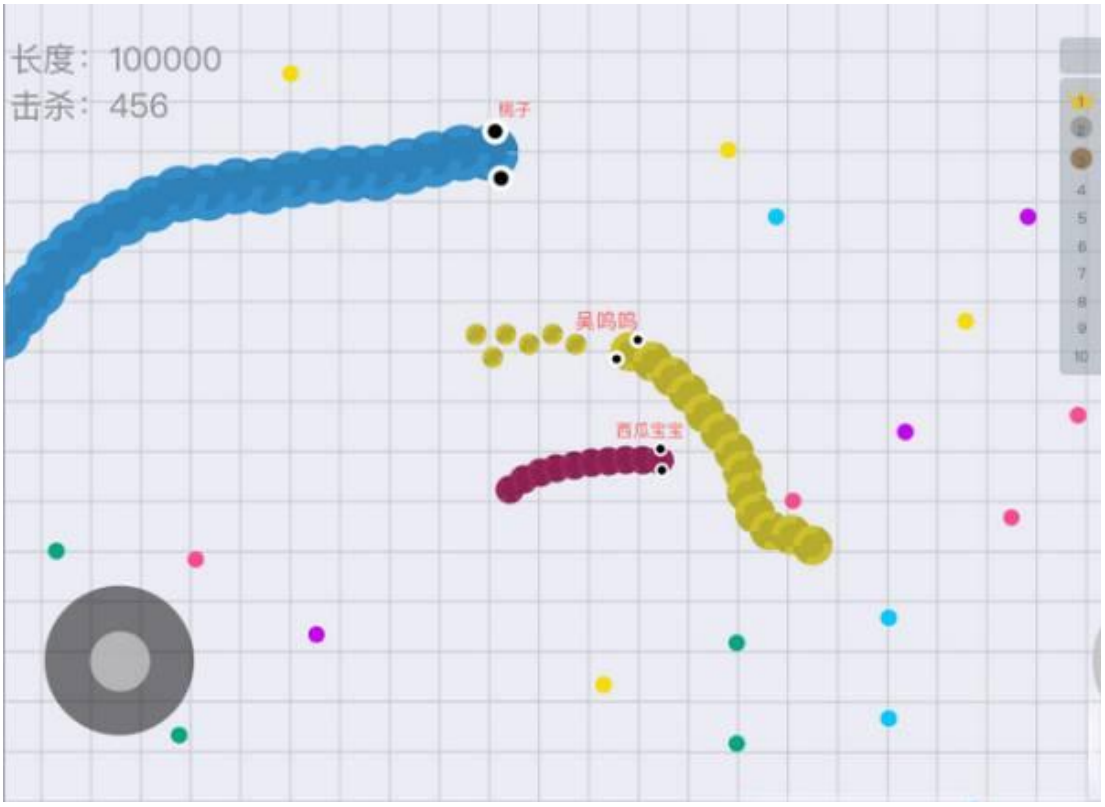
- Methods of organizing large amounts of data
    - Need to organize program data according to problem being solved

Data Structures: Why?

- Program design depends crucially on how data is structured for use by the program

- Example:
  - a sorted list of integers stored in an array
  - searching for specified items, print the data in desired order, or modify the value of any data item

Data Structures: Why?



Data Structures: Why?

广州地铁线网示意图
Guangzhou Metro System Map
广州地铁 全程为你
-96891
www.gzmtr.com

Algorithm Analysis: Why?

- Correctness:
  - Does the algorithm do what is intended.
- Performance:
  - What is the running time of the algorithm.
  - How much storage does it consume.
- Different algorithms may correctly solve a given task
  - Which should I use?

Terminology(I)

- **Abstract Data Type** (ADT)
  - Mathematical description of an object with set of operations on the object. Useful building block.
- Algorithm
  - A high level, language independent, description of a step-by-step process
- Data structure
  - A specific family of algorithms for implementing an abstract data type.
- Implementation of data structure
  - A specific implementation in a specific language

Terminologies (II)

- A **data structure** is the physical implementation for an ADT.
  - (C++) An ADT and its implementation together make up a **class**.
  - Each operation associated with the ADT is implemented by a **member function** or **method**.
  - The variables that define the space required by a data item are called **data members**.
  - An **object** is an instance of a class – created during the execution of a computer program.

Terminologies (III)

```
/* The ADT for a list */

template <typename E> class List { // List ADT
   // Clear contents from the list, to make it empty.
   virtual void clear() = 0;

   // Insert an element at the current location.
   // item: The element to be inserted
   virtual void insert(const E& item) = 0;

   // Remove and return the current element.
   // Return: the element that was removed.
   virtual E remove() = 0;

   // Move the current position one step right. No change
   // if already at end.
   virtual void next() = 0;

   // Return: The number of elements in the list.
   virtual int length() const = 0;

   // Return: The position of the current element.
   virtual int currPos() const = 0;
};
```

Algorithms vs Programs

- Proving correctness of an algorithm is very important
    - a well designed algorithm is guaranteed to work correctly and its performance can be estimated
- Proving correctness of a program (an implementation) is fraught with weird bugs
    - Abstract Data Types are a way to bridge the gap between mathematical algorithms and programs

Tips for Learning

- Practice, practice and practice
  - https://pintia.cn/ or the like
- Read textbook
- Complete homework independently
  - This is an exercise to test your knowledge and how much you learn
- Raise questions!
  - Do not delay your questions until exams
    - https://docs.qq.com/