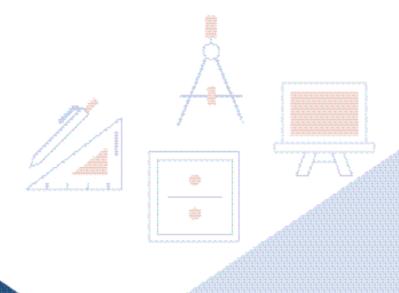




Data Structure

CS 240

Week 1 豌豆射手



SavvyUni CS 240 FAQ by 豌豆射手



关于我?

- UW 4A CS AI Option 在读
- · 连续三个学期没学期上(等效)7门课
- · CS 341 Algorithm: 99
- 上个 Coop Term 在一家保险公司担任 Full Stack Developer

为什么要上 CS 240?

- 为了学习更多解决问题的方法
- 为了面试
- 为了毕业

来参加这节课的目的是什么?

- 为了分!
- 学知识!
- 用尽可能少的时间拿尽可能高的分!
- 用尽可能少的时间拿尽可能高的分的同时还把知识学会!

CS 240 Overview

Data Structures and Data Management

Course Description

Introduction to widely used and effective methods of data organization, focusing on data structures, their algorithms, and the performance of these algorithms.

Prerequisite: CS 245, CS 246, Stat 230

- · CS 136: Tree, Stack, Queue, Dictionary, Sorting, and Big-O Notation
- ・ CS 245: Loop Invariant (并不会直接用到,但是会帮助你理解)
- · CS 246: 用 C++ 写代码
- Math 137: Limit, l'Hôpital (洛必达)
- Stat 230: Probability, Expectation, and Binomial Distribution

Topic

· Asymptotic Analysis (Formal Definition of Big-O Notation)

Data Structure (Extended from CS 136)

• Queue → Priority Queue and Heap

ullet Sorting ullet Selection, Radix Sort, and Interpolation Search

BST → AVL-Tree and B-Tree

• Dictionary(map) → Tries, Skip Lists, Ordering, and Hashing

Algorithm

- Range Search
- String Matching
- Data Compression

Marks Breakdown

Written Assignment 40%Programming 5%Mini Test 20%

Midterm Assessment 10% Oct 26 - Oct 27

Final Assessment 25% TBD

除了 Asymptotic Analysis, 其他章节相互独立

Terminology

Problem Desi	red /	rel	ationship		
• Example: S	Sorting Probl	em 排序问题			
	equence of				
• Output: A	reordering				
Instance of a	Problem A	n sati	sfy all		
Solution of a	Problem	for spe	ecified		
Size of an ins	stance A		that meas	sure of the size	e of the instance
Example:					
 For sorting 	g problem, th	ie size is defii	ned to be th	ne of t	he input array
		-	ibonacci N	umber), the	size is defined
to be		_			
Algorithm A		of	th	at transform _	to
Correctness output.	An algorithm	is said to be c	orrect if for _		, it halts correct
An incorrehalt with _	ct algorithm outp	=	on sor	ne input, or	
Program An		of an algorith	nm using a sp	pecial progran	nming language
Data Structu	re A way to _	and _	da	nta	
• No single I	Data Structur	e is			
•	tant to know				
Pseudocode	An method o	f	an	to)

Pseudocode

Use most clear and concise to specify an algorithm

Sometime ______ ClearSometime _____ ClearAvoid _____ ConciseAvoid _____ Concise

伪代码是给人看的,所以 每个人写的伪代码可能不 太一样,会有自己的特点

Some Conventions

while, if-else has the same meaning in C, C++

for has different meaning in C, C++. It is flexible, for example

- for i = 0 to n 1: print(i)
- for x in A. (Similar to for (auto x: vector))

Indentation indicates block structure

// 你看不见这一行,因为 // 后面跟的是注释

Array indexing is similar in C, C++

•

Variable refers array or object are passed by reference

• If an variable refer to _____, we give it special value _____

其他习惯问题

- · 有的人用← 当 assignment,本讲义里用 =
- ·有的人用=做 equality testing,本讲义里用==

Example: Binary Searh in Array

Iterative version Binary Search in C Language

```
int binary_search(const int arr[], int start, int end,
1.
2.
                        int key) {
3.
         int ret = -1;
4.
         int mid;
5.
        while (start <= end) {</pre>
            mid = (start + end) / 2;
6.
7.
            if (arr[mid] < key) {</pre>
8.
                start = mid + 1;
9.
            } else if (arr[mid] > key) {
                end = mid - 1;
10.
            } else {
11.
12.
                ret = mid;
13.
                break;
14.
            }
15.
16.
        return ret
17. }
```

Iterative version Binary Search Pseudocode

```
BinarySearch(A, start, end, key)

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.
```

Analyze an algorithm

In CS 240, Analyze an algorithm means

- Analyze the _____ of given algorithm
- Analyze the _____ of given algorithm

Step to Analyze Running time of Algorithm

- 1. Express algorithm in _____
- 2. Identify _____ in pseudocode
- 3. Express running time as a ______ that _____ primitive operations where ___ is the input size.
- 4. Determine _____ of ____ with _____

Example of Primitive Operations

Arithemetic

- Addition
- Subtraction
- Multiplication
- Division
- Remainder
- Floor
- Ceiling

Assignment

Control Flow

- If-else
- For, while
- Function Call
- Return

Array Indexing

Example:

Category A - For loop (Easy)

```
Sum(A)
A:array of size n
1.  i = 0
2.  acc = 0
3.  for i = 0 to n - 1
4.   acc += A[i]
5.  return acc
```

Category B: Recursion (Easy)

```
BinarySearch(A, start, end, key)
    if (start > end)
1.
2.
        return "Did not found"
3.
    mid = (start + end) / 2
    if (A[mid] > key)
4.
5.
        return BinarySearch(A, start, mid - 1, key)
6.
    else if (Arr[mid] < key)</pre>
7.
        return BinarySearch(arr, mid + 1, end, key)
8.
    else:
9.
        return mid
```

Category C: While - loop (Hard)

```
BinarySearch(A, start, end, key)
1.
     while (start < end)</pre>
2.
         mid = (start + end) / 2
3.
         if (A[mid] > key)
4.
             start = mid + 1
5.
         else if (A[mid] < key)</pre>
             end = mid - 1
6.
7.
         else
8.
             return mid
     return "Did not found"
9.
```

Exercise

For loop 1

```
    m = 0
    for i = 1 to 3n
    m = m * 4
    for j = 1025 to 2048
    for k = 4i + 1 to 6i
    m = m + 4 * k
```

For loop 2 (give an upper bound)

```
    x = 0
    for i = 1 to floor(log n))
    for j = 1 to 2<sup>n</sup>i
    x = x + i * j
```

Exercise

while loop 1

2.
$$j = 0$$

3.
$$k = 1$$

4.
$$while(2*j + k \le n)$$

$$5. \qquad i = i + j$$

5.
$$i = i + 1$$

6. $j = j + i$
7. $k = k + 1$

7.
$$k = k + 1$$

iter	t	i	j	k	2j+k

Nested loop (give an upper bound)

1.
$$x = 0$$

2. for
$$i = 1$$
 to n

$$j = i$$

5.
$$i = i / 2$$

Asymptotic Notation

Bi	g-	\bigcirc	N	0	ta	ti	\mathbf{O}	n
_	0				-	٠.	_	

Big-O Notation
Meaning
 O(g(n)) is a

• It is a set of function whose growth rate is _____ or ____ than g

• If ______ of ____ of ____

Formal Definition ______if _____constants _____ and _____ such that ______ for all _____

Example

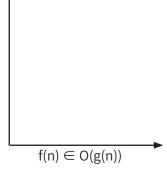
Proof _____ from first principle

Proof _____ from first principle

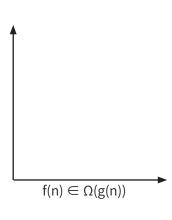
Proof _____ from first principle

当输入规模无限增加时, 算法的运行时间如何随着 输入规模的变大而增加?

有时候我们也会用 f(n) = O(g(n)) 代替 $f(n) \in O(g(n))$ 来方便递归 计算



Definition 有时也叫 First Principle



 $f(n) \in \Theta(g(n))$

Big-Ω Notation

Meaning

- $\Omega(g(n))$ is a set of function whose growth rate is ____ or ___ than g
- If ______ of ____ of ____

Formal Definition ______if _____ constants _____ and _____ such that ______ for all _____

Example

Proof _____ from first principle

Big-Θ Notation

Meaning

- $\Theta(g(n))$ is a set of function whose growth rate is _____ to g
- If ______ of ____ of ____

Formal Definition ______if ______constants _____ and _____ such that ______ for all _____.

Example

Proof _____ from first principle

Example		
	from first principle	
Duant	form first main sinks	
Proot	from first principle	

Example	
Example Proof	from first principle
Proof	from first principle
11001	. Hom mat principle

Summary

- Basic Concept about Data Structure & Algorithm
- Pseudocode
- · How to analyze an algorithm
- Asymptotic Notation (O, Ω , Θ)

Three level in Mastering CS 240

- Known basic concept and able to directly apply algorithm.
- Find property of DS/Algortihm with given input, or find specific input that satisfy some requirement.
- Analyze variation of DS/Algorithm and design new DS/Algorithm.



小助手微信



微信公众号



课程服务咨询电话

- +1(647)926-9109
- +1(226)978-6660



有爱有科技 || ||有你有未来