CS2010 – Data Structures and Algorithms II

Lecture 01 –Introduction

stevenhalim@gmail.com

National University of Singapore

School of Computing

Welcome ☺

- Teaching Staffs Introduction
  - See IVLE Workbin for special PPT
- Class Ratio
  - Tutorial: 10:118 = 1:~12
    - Class participation is a must! (5%)
  - Lab: 7:118 = 1:~17
    - Demo attendance (5%)
    - Take home Problem Sets (15%)

Outline

- What are we going to learn in this lecture?
  - Some admin stuffs (a quick one)
    - Clicker distribution + loan form signing (with help of TAs)
    - Game system and the rules for PSes
    - CP2.5 book sales (only for those who are interested)
  - CS1020 Quick Review (First serious try of clickers system)
    - And comparison with CS2010
  - Problem Solving Paradigms
    - Complete Search
    - Divide and Conquer
    - Greedy
    - Dynamic Programming

Course Information

- By viewing CS2010 IVLE, you will be able to know:
  - The complete class schedule
  - Important dates (esp Saturday help sessions + quizzes)
  - Teaching modes, Assessment/Grading scheme
  - FAQ
  - Lab and Tutorial Assignments
- If you have not read them… do it **NOW**
- In the next few slides,
  I will only highlight a few more important stuffs

CS2010 versus CS2020: The Similarities

- *Nearly* half of the syllabus are the same (CS2020 also has to cover CS1020 syllabus)
- Both modules use the clicker system http://www.cit.nus.edu.sg/classroom-response-system/ (very positive feedback in recent semesters)
- Both are almost two years old by now
  - CS2020 starts Jan 2011
  - CS2010 starts Aug 2011

CS2010 versus CS2020: The Differences

CS2020

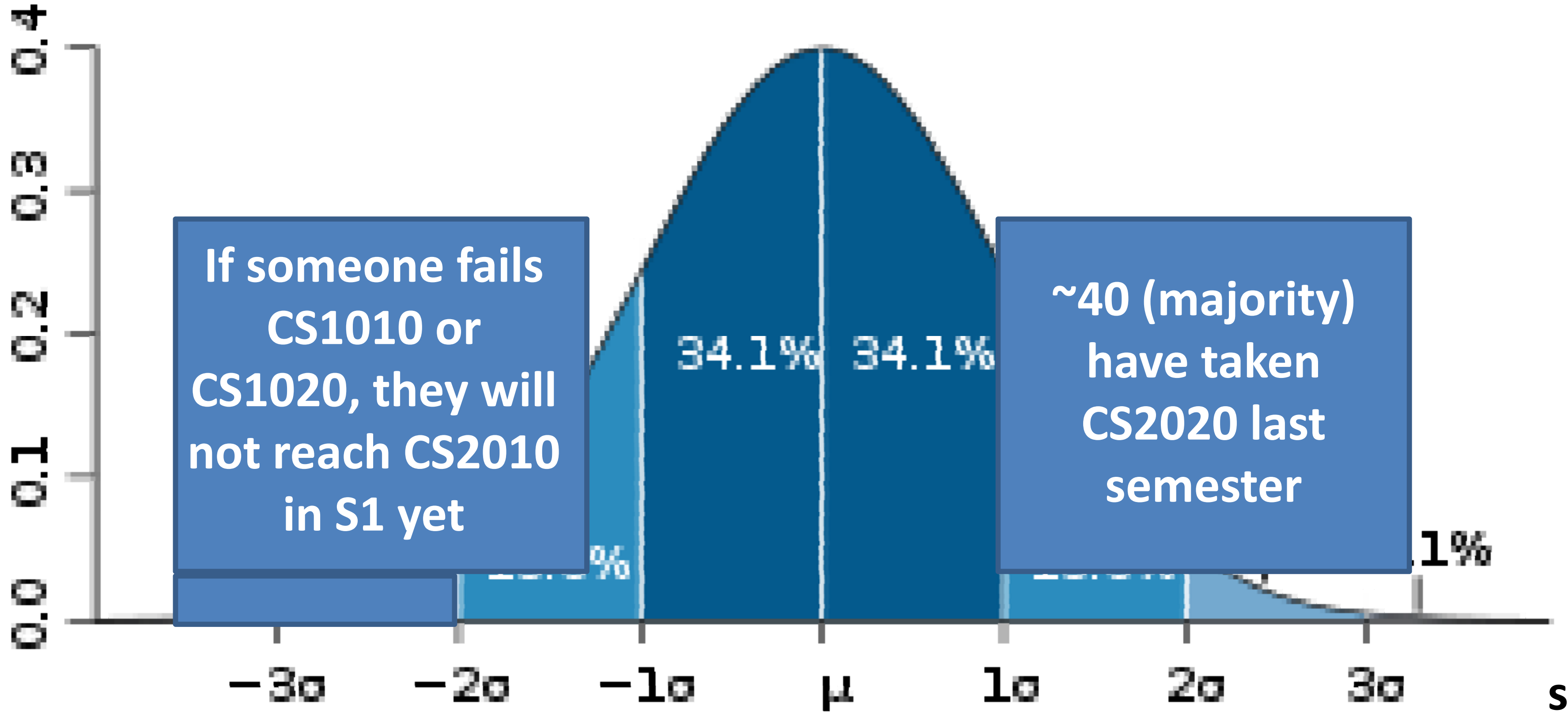
- Speed →→→
- Workload ↑↑↑
- Level of difficulty ↑↑↑

CS2010

- Speed →
- Workload → or → → →
  (your choice)*
- Level of difficulty
  → or → → →
  (also your choice)*

* to be explained

Typical Class Profile (CS2010 in S1)



If someone fails CS1010 or CS1020, they will not reach CS2010 in S1 yet

~40 (majority) have taken CS2020 last semester

~300 SoC students

And from these remaining students, about half choose IS and do not need to take CS2010 in S1

How to get B-/B grade in CS2010

- Simple, just do the minimum requirements
- The level of difficulty of CS2010 in S1 is calibrated for this group of students
  - Just by *doing OK* in all graded components will give you B-/B

How to get B+/A- grade in CS2010

- Do all those required for B-/B grade
- Each graded components are set with 75:100 rule
  - 75% marks are doable for students who aim to get B-/B
    - Last year was 85:100 (a bit too easy)
- Improve your mathematics proficiency
  - One indicator is your performance on CS1231*
- So, if you occasionally spend some time to do hard questions in weekly PS/bonus PS (can take hours…) + occasionally able to solve the harder questions in Quiz 1/2/Final, then you will be in the B+/A- range

How to get A/A+ grade in CS2010

- Do everything that is graded… with near perfect score…
- Be in the leader board of http://cs2010.ddns.comp.nus.edu.sg
- Proficiency in discrete structures (graphs, trees) and proofs
- Learn more beyond CS2010 syllabus by doing the extra exercises at every tutorial, PS, and Quiz 1/2/Final
  - Advertisement: buy Competitive Programming book 2.5nd ed
  - Then use that extra knowledge to solve Steven's tricky/hard quiz 1/2/final exam questions which can be expected to appear at maximum 20-25% of the weightage of a certain test
- A/A+ students in CS2010 are invited to take Steven's CS3233 course in Sem2 AY 2012/2013 and/or to be Steven's TA for next year's CS2010 ☺

For those who struggle with CS1010/CS1020/CS1231 series

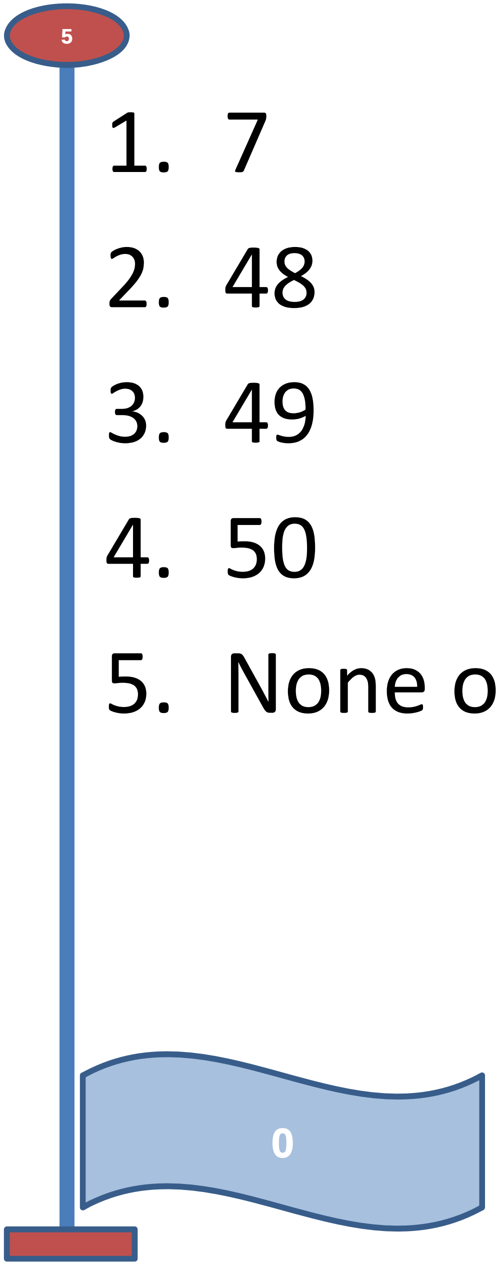
- Come to Steven personally and I will try to make some extra arrangements for you
  - Several Saturday help sessions are in the pipeline
- Note that you are now at University level
  - I will not beg you to come to me,
    but will only help you if you approach me
  - The earlier the better…

CS2010R

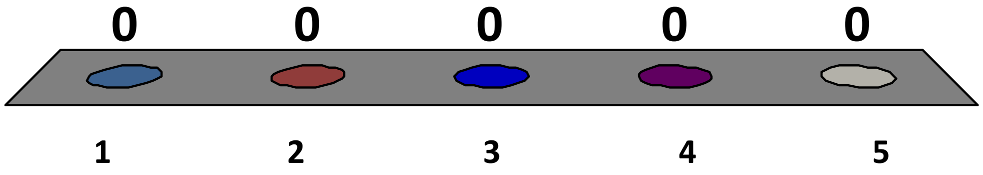
- New for this semester
  - Eligible for ex CS2010/CS2020 or current CS2010 students
  - Not appropriate for those who have taken CS3233
  - Case by case basis for other interested applicants
- Extra 1 MC
  - Extra 1 MC = extra 3 hours/week, or extra 39 hours/sem
- To do:
  - Have to do PS1R, PS2R, PS3R, PSBonus, PS7R
  - Graded by Lab TA, but (see below)
  - Have to meet Steven once after PS7R to present solutions/finalize the marks

THE CLICKERS…

What is the answer of 7*7?



1. 7
2. 48
3. 49
4. 50
5. None of the above



About the Clickers

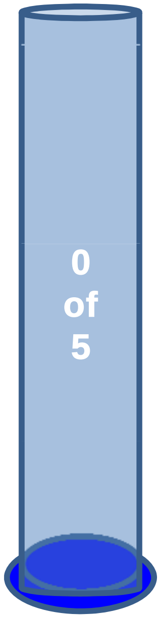
- We will use this useful gadget for all CS2010 lectures and *maybe* also during tutorials
  - http://www.cit.nus.edu.sg/classroom-response-system/
- I have 70 clickers for more than half of you (118)
  - However, you can choose not to loan it
  - Because to get one clicker for yourself this semester, you have to **sign a "loan form"** that tells me that you are willing to *bring the clicker for all our lectures (and tutorials)*, and *if you lose/broke it*, you are willing to pay a replacement fee of **48.15 SGD/clicker**

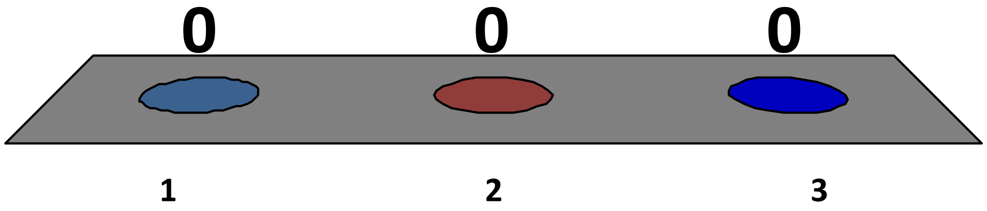
Distribution of Clickers

- We will employ “parallel processing”
- Those who are willing to take responsibility of loaning this gadget for **one semester** will:
  - Come forward with your matriculation card
  - Meet one of the TA (each TA carries some clickers)
  - Sign at the appropriate slot in the loan form
  - Take one clicker, guard it throughout 1 semester
  - Return these clickers back to me during the **last lecture** on Week12 (no class on Tue of Week13)

Now, let’s try. You are: (only last vote counts)

1. A male student
2. A female student
3. None of the above





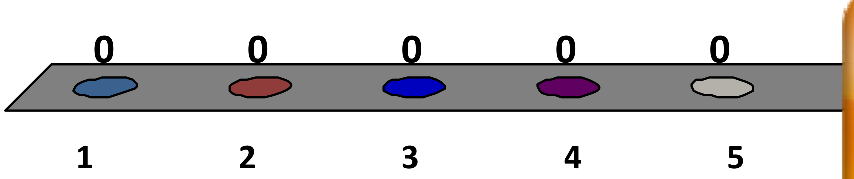
What is the answer of 7*7?

(You can select up to two answers/clicker)

That is, the last two votes count

1. 7
2. 48
3. 49
4. 50
5. sqrt(2401)
6. None of the above

0 of
5



10

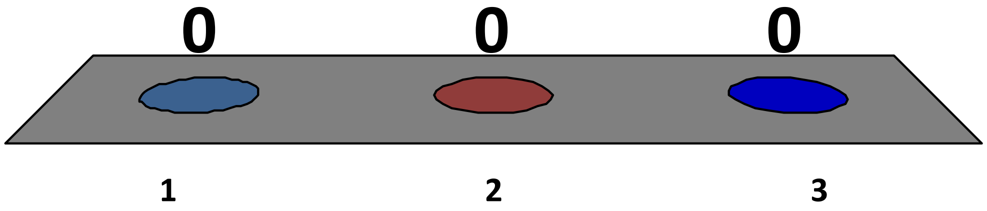
Countdown
Timer
On Slide

Game System for Weekly PS

- Our Game System URL:
  - http://cs2010.ddns.comp.nus.edu.sg
- Register with Facebook connect
  - Sign up for a free Facebook account if you do not have one yet
    - Then you can add each other’s Facebook account
    - I also have one, you can add me if you want
    - We also have CS2010 Facebook group, join it!
      - https://www.facebook.com/groups/241724769269875/
- Now, let’s use clicker again to do a quick survey
  - See how useful this gadget is ☺

I…

1. Have registered at CS2010 game system
2. Have Facebook account but have not registered at CS2010 game system
3. Do not have Facebook account yet ☹



Weekly PS (1)

- There will be 8 Problem Sets (PSes) in CS2010
  - 7 of them are “babies related” ☺, chronological, and use real-life examples (OK, some are a bit exaggerated)
  - Each 8 normal PSes has similar weightage
  - There will also be 1 bonus PS (smaller weightage)
    - Released at the middle of the semester
    - Useful to catch up if you lose some marks in some early PSes or as buffer for some future PSes
  - More details in the next few slides

Weekly PS (2)

- Rules for Weekly PS:
  - Collaboration is strongly encouraged, but you have to write the solution (i.e. the Java codes) **by yourself**!
  - Posting algorithm/data structure ideas to IVLE discussion forum is strongly encouraged, but you can **never** upload your Java codes to discussion forum **before deadline**, even if it is a “buggy” one! You have to write and debug **your own** Java codes!
    - Severe penalty for those who breach this rule

Weekly PS (3)

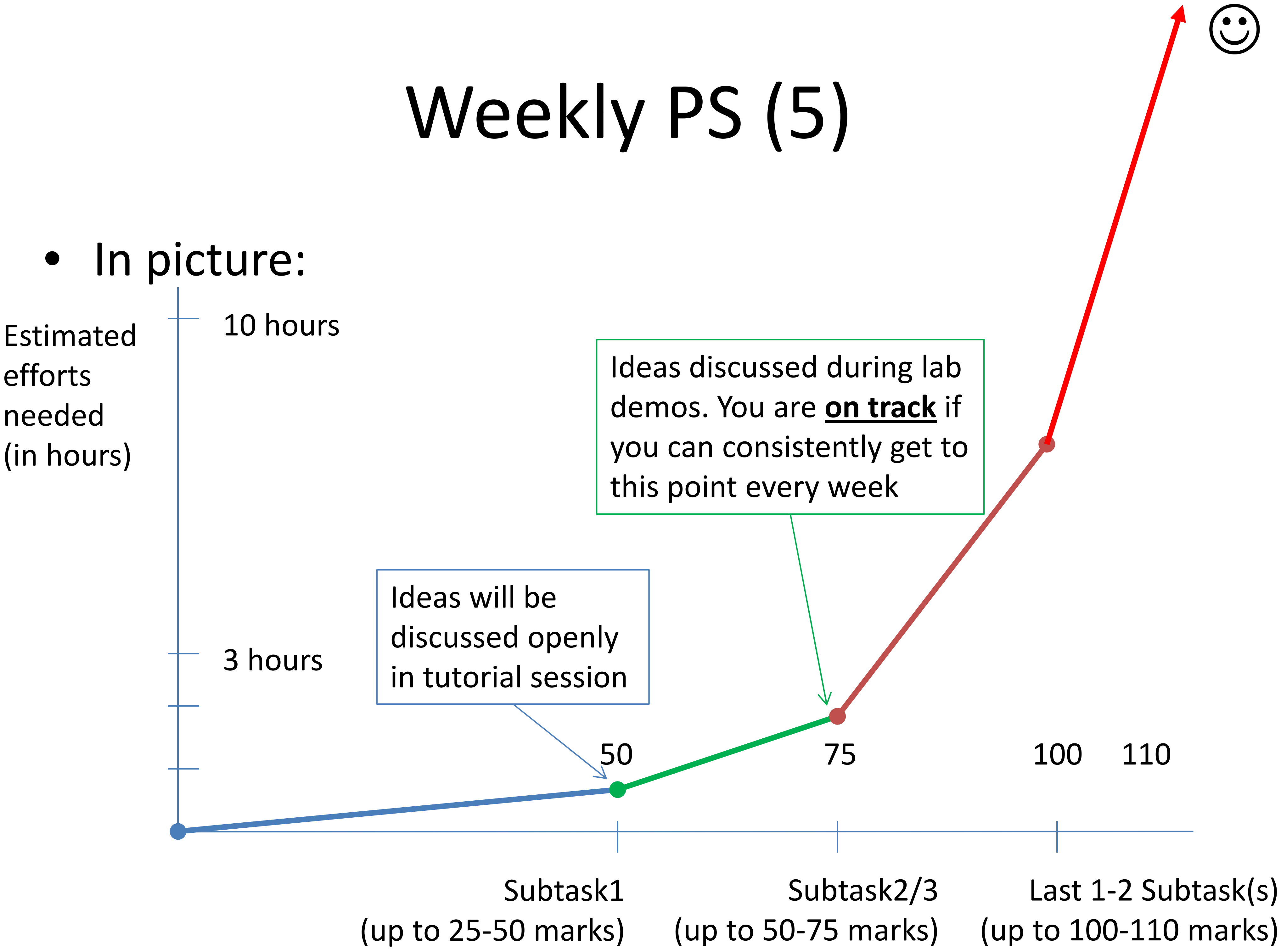
- PSes are the core of CS2010
  - Most likely you will spend many hours (depending on your aim) discussing and implementing the solutions
    - It is designed as one “simple” problem with “subtasks” with *gradual* level of difficulty
      - The easier subtask just require CS1010-CS1020 knowledge
      - Most subtasks definitely require CS2010 knowledge
      - The last subtask (usually the R-option) require perhaps CS3230/CS3233 knowledge++
  - The ideas (not the Java codes) that can solve Subtask 1-2 of each PS will be discussed during the tutorial sessions ☺
    - So, you can score up to 25-50 (out of 100) marks by understanding what is discussed in the tutorial and then implementing it

Weekly PS (4)

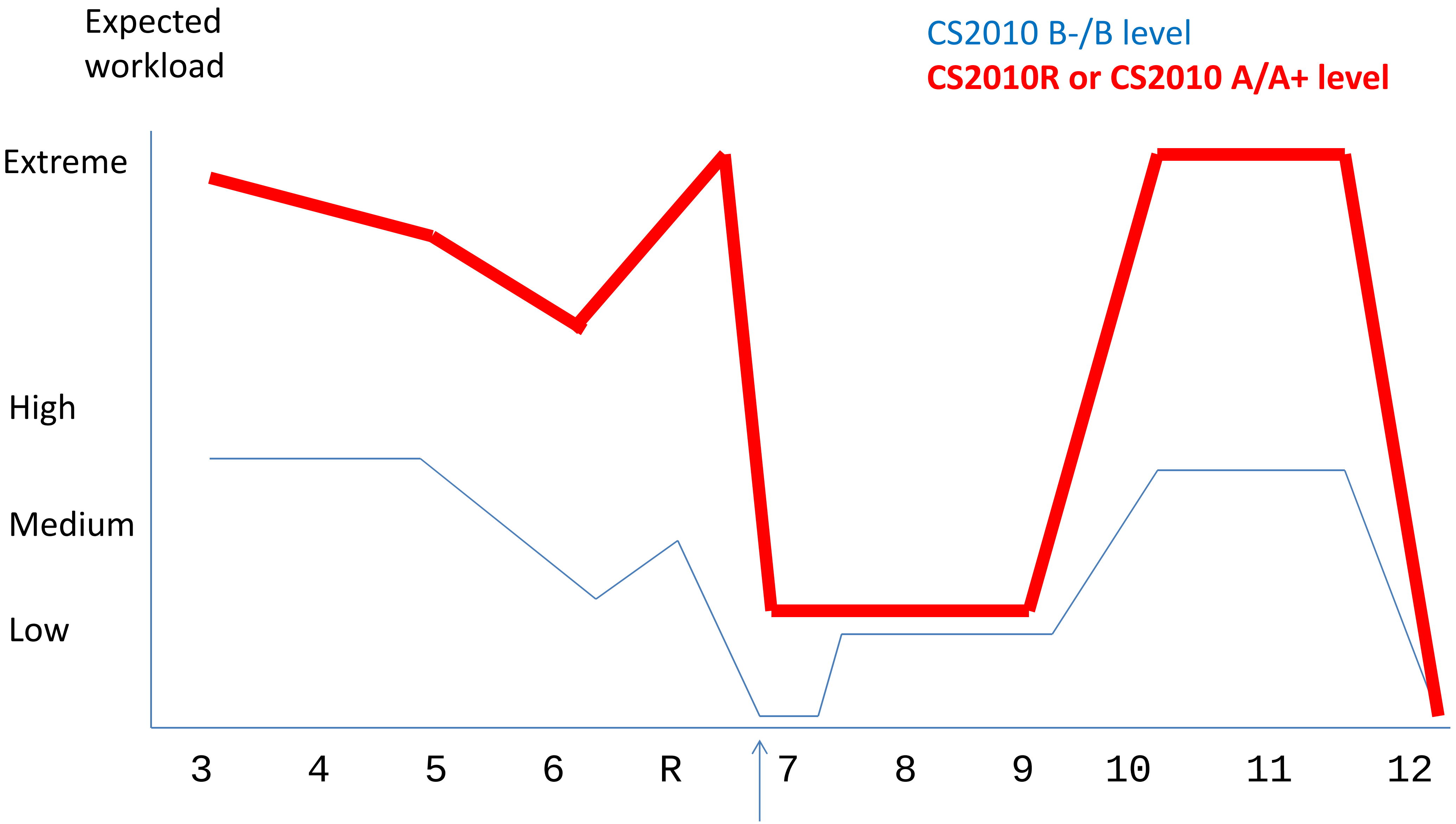
- PSes are the core of CS2010 (continued)
  - The implementation of the required technique (but not the actual solution) that can be used to solve parts of the harder subtasks will be discussed during the lab demos
    - You can score up to 50-75 (out of 100) marks by understanding what your Lab TA is trying to tell you during his/her lab demo
  - The last subtask (or the bonus subtask/R-option) of each PS is designed for those who are aiming to get A/A+ in this module… they are either ***difficult*** or ***tedious***
    - If this is your aim, you may end up spending *hours* to solve this
    - The majority of you do not have to attempt this every week :O

Weekly PS (5)

- In picture:

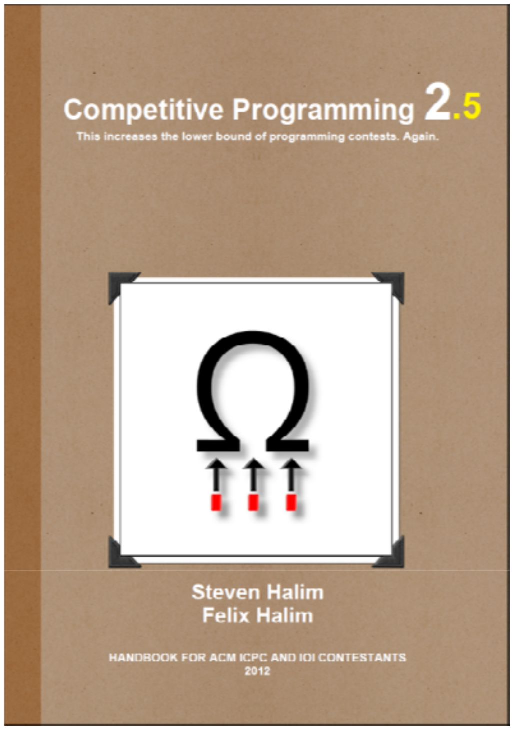


Weekly PS (6)



For your other modules midtests ☺

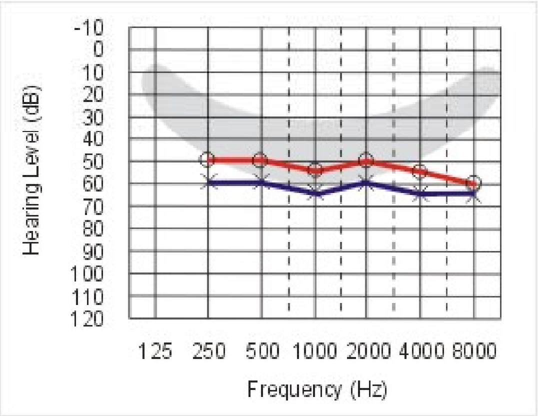
CP 2.5 Book Sales


Competitive Programming 2.5
This increases the lower bound of programming contests. Again.
Ω
Steven Halim
Felix Halim
HANDBOOK FOR ACM ICPC AND IOI CONTESTANTS
2012

- Not compulsory at all
  - This is after all a CS3233 text book
- Contains about “~50%” of my algorithmic knowledge so far
  - Definitely useful to tackle the last subtask of each PS and to answer some tricky questions during Quiz1/2/Final (maximum 20-25% per each test)
- If you are interested, come to me during lecture break (which is now) or after lecture
  - Local sales at **20 SGD/copy** (cheaper than CP2, 25 SGD, as chapter 5-9 are not inside CP2.5)
  - CP2.5 uses 186 pages to cover Ch1-4 whereas CP2 only uses 120 pages
  - I have 50 copies for Week01, first come first serve
- *News: 15 Aug 2011: This book is also in Central Library RBR (Reserved Books/Readings) QA76.6 Hal 2011*

About My Lecture Style

- Should be quite interactive with help of Clickers
  - Sometimes I will invite some students to the front of LT
- But if you need verbal clarifications during lecture…
  - Please wait until lecture breaks or after lecture
    - Reason: hearing issue… ☹



  - Alternative: Post questions in the IVLE discussion forum/FB

10 minutes break, and then…

A little refreshing game using our new gadget (the clickers)

CS1020 (AND CS1231 FOR MAJORITY OF YOU) REVIEW

CS1020 – OOP

- Object Oriented Programming (OOP)

```
class BankAccount {
  private int Balance;
  public BankAccount();
  public void Deposit(int Amount);
  public void Withdraw(int Amount);
  public int CheckBalance();
}
```

- We will use OOP principles in our CS2010 PSes
  - e.g. We use “IntegerPair” and “IntegerTriple” in CS2010

CS1020 –Algorithm Analysis

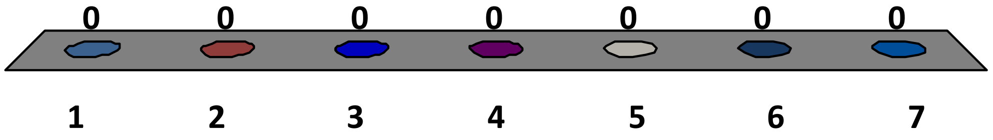
- Big O notation, the O(g(n)) stuffs
- In CS2010, we will
  - Extensively use this algorithm analysis, e.g.

    ```
    sum = 0;
    for (int i = 0; i < n; i++)
      sum += A[i];
    ```

    - is an O(n) algorithm
  - Use more advanced algorithm analysis skills for some cases

$$f(n) = 5000 \log n * \log n + 5n - n + \mathrm{sqrt}(n^2)$$
$$f(n) =$$

1. $O(n^2)$
2. $O(n \log \log n)$
3. $O(n \log n)$
4. $O(n)$
5. $O(\log \log n)$
6. $O(\log n)$
7. $O(1)$



0 of 5

CS1020 – Linear Data Structures (1)

- Data Structure is a way to store and organize data
  - We will frequently abbreviate it as **<u>DS</u>**
- A good DS is needed to support *efficient*:
  - Insertions: add a new item into the DS
  - Searches: is item X inside the DS or not?
  - Deletions: remove a certain item out from the DS
  - Queries: how many items is the DS?, what is the min item in the DS?
  - Updates: combination of (or a more efficient form than) "delete the old item" and "insert the new item"
- Different situations may require different DS

CS1020 – Linear Data Structures (2)

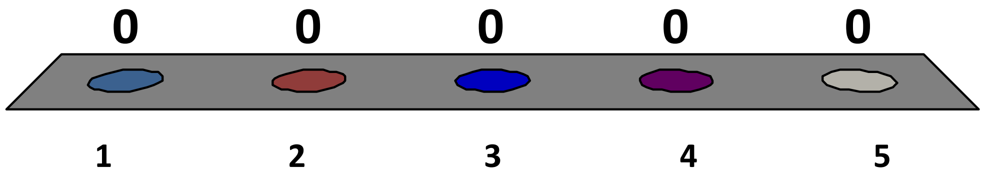
- Linear Data Structures that you learned in CS1020:
  - Items listed in left-to-right (or top-to-bottom) order
  - Array (fixed size)/Vector (resizeable),
  - Linked List
  - Stack: Last In First Out (LIFO)
  - Queue: First In First Out (FIFO)
- You will learn one more in CS2010:
  - Lightweight array of Boolean (Bit manipulation)
- Then, you will learn the non-linear DS:
  - Binary Search Tree, Heap (Priority Queue), Graph

Introducing…

- http://www.comp.nus.edu.sg/~stevenha/visualization
- A **visualization project** between myself, Zi Chun (3rd year SoC), Victor Loh (graduate, work @ FB), Felix (my brother, graduate, work @ Google) and 4 FYP students (Albert, Trang, Peter, Duy)
- A new way of learning data structures & algorithms
  - Explore them ON YOUR OWN!
- Now, if you have either: **iPhone** (or other HTML5 compatible smartphones), **iPad**, or **laptop**, visit that URL and follow me ☺
  - We will start with LinkedList/Stack/Queue visualization

What is the best time complexity to search for an item in an unsorted linked list of size N?

1. $O(N^2)$
2. O(N log N)
3. O(N)
4. O(log N)
5. O(1)

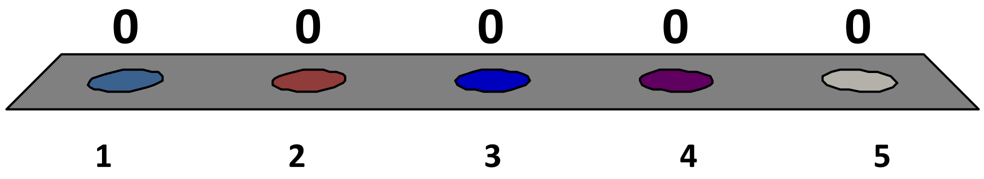


0 of 5

What is the best time complexity to search for an item in a sorted linked list of size N?

Hint: Binary Search?

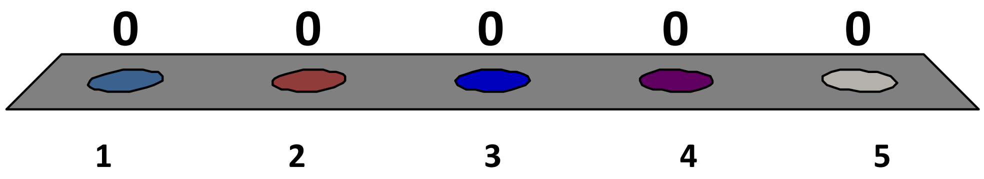
1. $O(N^2)$
2. O(N log N)
3. O(N)
4. O(log N)
5. O(1)



0 of 5

Four integers are inserted into a Stack one by one, then the top two are popped out, then the fifth integer is inserted into the same Stack. **Who is on top of the Stack now?**

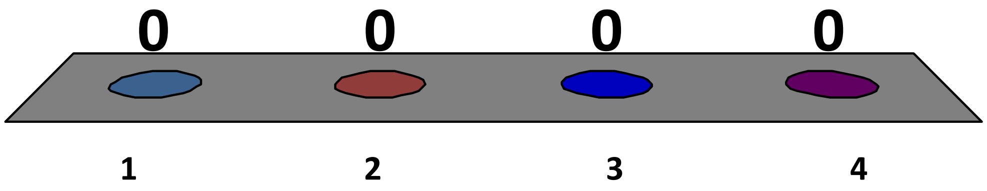
1. The first integer
2. The second integer
3. The third integer
4. The fourth integer
5. The fifth integer



0 of 5

Three person “Steven”, “Grace”, “Felix” entered a queue, in that order. After waiting for a few minutes, the person in the front of the queue is called. **Who is he/she?**

1. Steven
2. Grace
3. Felix
4. Someone else



0 of 5

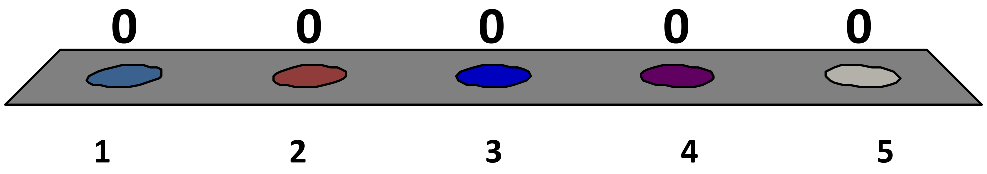
CS1020 – Sorting

- What you learn in CS1020:
  - O(N^2) Selection Sort, Bubble sort, Insertion sort
  - O(N log N) Merge sort
  - Expected O(N log N) Quick sort if the pivot is randomized
    - Can go to O(N^2) otherwise (but this is what you learned in CS1020)
- In CS2010:
  - If not explicitly stated, you can use Java library functions, e.g. Collections.sort for all your sorting needs
  - We will learn more sorting algorithms: BST Sort, Heap Sort

What is the best sorting algorithm to sort this **almost sorted** sequence?

X = { 1, 1, 1, 1, 1, 3, 3, 4, 5, 6, 7, 2, 1M}

1. Selection Sort
2. Bubble Sort
3. Insertion Sort
4. Quick Sort
5. Merge Sort



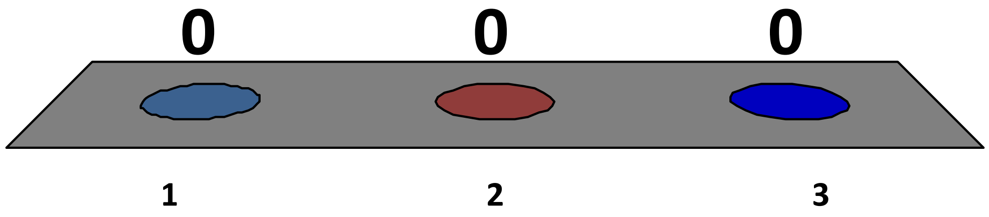
0 of 5

CS1020 – Recursion

- In CS1020, you may have learned these examples:
  - “Countdown”
  - Factorial
  - Printing a linked list in reverse order
  - Towers of Hanoi
  - N choose K
  - Recursive binary search
  - Fibonacci
- In CS2010, we will see *much more* recursion

I…

1. Have no problem with recursion examples shown in CS1020
2. Am lost with recursion
3. Am very lost with recursion ☹



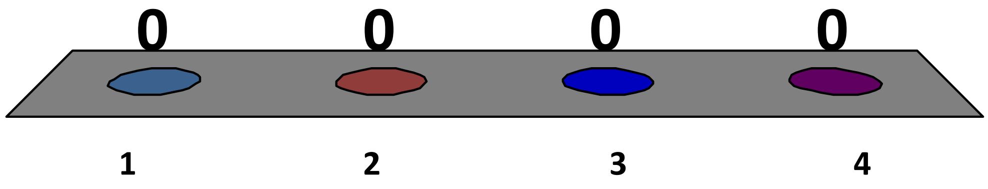
0 of 5

CS1020 – Hashing

- Concepts that you learn:
  - Direct Addressing Table
  - Creating good Hash Function
  - Handling collisions: Birthday paradox
    - Separate chaining
    - Linear probing, quadratic probing, double hashing

Give me a pair of numbers x1 and x2 so that
h(x1) = h(x2) for h(x) = (x * x) % 7
(you can select up to 4 options)

1. x1 = 71, x2 = 55
2. x1 = 77, x2 = 66
3. x1 = 7, x2 = 15
4. x2 = 9, x3 = 147



0 of 5

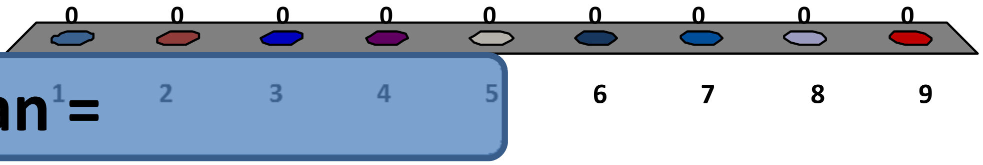
CS1231 – Discrete Structures

- Another relevant module for CS2010 is CS1231
- But CS1231 is not a pre-req of CS2010
- Relevant stuffs are:
  - Discrete structures: Graphs and Trees
  - Proofs, we will see lots of them (simpler form)
- In CS2010, we will see all these discrete structures practically throughout the semester
  - That's it, lots of trees and graphs and proofs (simpler form)
- Let's see the profile of CS2010 students

My CS1231 (or MA1100) grade

(don’t be shy, this is anonymous)

1. A+
2. A
3. A-
4. B+
5. B
6. B-
7. C+/C
8. D+/D/or F
9. Haven’t take that mod!



0 of 5

Mean =

Source:

-. Competitive Programming 2.5, Chapter 3 (overview)

-. Introduction to Algorithms, 2nd ed, Chapter 7 and 15-16

This is what we will learn throughout in CS2010 ☺

PROBLEM SOLVING PARADIGMS

Complete Search

- Given an integer array A = {10, 7, 3, 5, 8, 2, 9}, n = 7
- Find the largest and the smallest element of A!

Divide and Conquer

- Given an integer array A = {10, 7, 3, 5, 8, 2, 9, …}, but now n = 100000 items
- What is the 12345th smallest item in A?
  - Can we use the previous Complete Search algorithm?

Greedy

- Given an integer array A = {10, 7, 3, 5, 8, 2, 9, …}, n is still 100000 items
- Find the largest gap *g* such that *x, y* ∈ *A* and *g* = |*x* − *y*|

Dynamic Programming

- Given an integer array A = {10, 7, 3, 5, 8, 2, 9, …}, but now n = 1000 items
- What is the **longest subsequence** of A that if viewed from left to right is always non decreasing?
  - {3, 5, 8} is a subsequence, and non decreasing
  - {3, 5, 8, 2} is also a subsequence, but 8 → 2 is decreasing
  - {3, 5, 8, 9} is the longest so far (ignoring the …)

In the Context of CS2010 (1)

- Lecture 2-3
  - **Divide and Conquer** principle in Data Structure
  - Binary Search Tree (BST)
  - Balanced BST: Adelson-Velskii Landis (AVL) Tree
- Lecture 4
  - Another **Divide and Conquer** principle in DS
  - Heap DS
  - Additional stuffs: Heap sort
  - **(Quiz 1 is up to here)**

In the Context of CS2010 (2)

- Lecture 5: Graph Basics
  - Graph DS & Traversal (revisit CS1231 material)
  - Depends on application: Complete Search or others
- Lecture 6: Minimum Spanning Tree (MST)
  - Greedy
- Lecture 7+8: Mid-semester Check-up (a review) and then Single-Source Shortest Paths (SSSP)
  - Bellman Ford’s, Dynamic Programming
  - Dijkstra’s, Greedy
  - **(Quiz 2 is up to here)**

In the Context of CS2010 (3)

- Lecture 9: Algorithms on DAG
  - Dynamic Programming
- Lecture 10: Algorithms on (Implicit) DAG
  - Dynamic Programming
- Lecture 11: All-Pairs Shortest Paths (Final up to here)
  - Dynamic Programming
- Lecture 12: Mystery Lecture
  - Let it be a mysterious one for now ☺
- Lecture 13: No class (Deepavali/public holiday)

That’s all for today

- We will gear up for the main topic of CS2010
  - Binary Search Tree (BST)
  - To do list at home:
    - For majority of you who have taken CS1231, this is a bit of revision of your CS1231 knowledge about trees ☺
    - For the rest, explore the new set of new lecture notes