

Midsem Preparation

Format
Summaries
S1 Debrief
Tips

Test Format

- Short answer questions (not limited to..)
 - State the correct answer/fill in the blanks
 - True/False + Justification
 - Implications
 - Create testcases
- Analysing / designing algorithms
 - Pseudocode + C++ code mix
 - “Programming essay”
 - Manually graded

Test Format

- Don't be afraid to ask questions during the test
- If you are stuck (for a while), clarify to ask if you are understanding the problem correctly
 - Examiner might forget to state common assumptions clearly when setting the questions
 - *Optimized* bubble sort or *standard* bubble sort?
 - Sometimes, the way examiners answer your queries will hint at the angle to approach the question

Technical Content

- Complexity Analysis
- Sorting Algorithms
- Sorting Applications
 - Binary Search
 - 2 Pointer Method
- Linear Data Structures
 - Includes implementation (using STL or not)

Soft Skills

- ~~Proving techniques~~ *not really expected in S2 midterm test yet*
- Reading code
 - C++ classes syntax
- Experience with STL Containers (Lab)
- Exam skills
 - Time management
 - **Use pencil**
 - Pseudocode or C++ code or **Mix**

Clarification

Q: “What is the level of pseudo-code that is acceptable?”

A: If I pass it to a CS1010 student, he/she should be able to write the code out **without ambiguity**.

Not acceptable: “Loop through the array” (*which part?*)

Acceptable: “Loop through the array from 0 to **N-1**”

Summary of C++ STL Containers

Container	Vector		Stack Queue		Deque		List*		Priority Queue	
Insert	push_back	O(1)	push	O(1)	push_back push_front	O(1)	push_back push_front	O(1)	push	O(logN)
Delete	pop_back	O(1)	pop	O(1)	pop_front pop_back	O(1)	pop_front pop_back	O(1)	pop	O(logN)
Random Access	[] operator	O(1)	NIL		[] operator	O(1)	Loop Through	O(N)	NIL	
Access	front back	O(1)	s.top q.front q.back	O(1)	front back	O(1)	front back	O(1)	top	O(1)
Sorted (default)	No (Use STL sort)		No		No (Use STL sort)		No (Use list.sort())		Yes	
Binary Search	lower_bound upper_bound	O(logN)	NIL		lower_bound upper_bound	O(logN)	NIL		NIL	
Iterators	Yes		No		Yes		Yes		No	

Note*: List can also insert and delete in O(1) at any iterator position.

Credit: SG NOI/IOI Training Team

Summary of common ADT Operations

ADT	Linked List	Stack	Queue	Doubly Ended Queue (Deque)	Priority Queue
Insert	insert(index, value)	push(value)	enqueue(value)	enqueueFront(value) enqueueBack(value)	enqueue(value)
Delete	delete(index)	pop()	dequeue()	dequeueFront() dequeueBack()	dequeueMax()
Access	get(index)	top()	front() back()	front() back()	top()
Search	search(value)	NIL	NIL	NIL	NIL
Ordering	As inserted	First-In-Last-Out (FILO)	First-In-First-Out (FIFO)	As inserted	Largest Priority First

Recall an ADT is independent of its implementation.

Worth printing?

- Lecture notes
- Tutorial slides (selectively?)
- Codes for problems you have solved

Use *algorithms* to order your notes.

Tips

Tips

Write the code in the most *natural* way first.

If you need to spend time to 'refer' to write C++ code:

- Write in pseudocode first
- Convert later when you have time

Main focus is still on *algorithmic thinking* and *application of data structures*.

Tips

Don't leave anything blank.

A *slow* algorithm is still worth partial marks :)

Be careful of crucial words (min/max/first/last/second)

Remember to calculate time complexity.

(Some seniors have lost 5 marks due to this)