Lecture 12

Wrapping Up

Lecture Outline

- PE2 Debrief
- Review
- Past exam questions
- Exam matters

PE2 Debrief

Discussions of the solutions
and preliminary grading remarks…

Module Review

CS1020E Objectives (1/2)

- Give an introduction to OO Programming (OOP) Model using C++ programming language, linear data structures, and algorithms for constructing efficient programs
- Emphasize data abstraction in program development (through ADTs)
- Emphasize efficient implementations of chosen data structures and algorithms

CS1020E Objectives (2/2)

- Linear data structures include arrays, lists, stacks, queues, and one Non-linear data structure: hash tables; together with their operations (insert, delete, find, and update).
- Recursion as problem-solving approach, for formulation of solution, and for programming
- Elementary analysis of algorithms
- Simple search and sorting algorithms, and divide-and-conquer problem-solving approach

Topics (1/2)

1. Basic of C++
2. OOP (I) – delivered by A/P Tan Sun Teck
3. OOP (II), Lab1 – Round the Ring
4. Useful Features in C++, Lab2 - Rectangles
5. ADT, ADT List, That Lab3 - Distributor
6. -Public Holiday- Lab3 cont+PE1, i.e. Lab3'

Recess Week

7. Linked List and Its Variations, Lab4 - TextEd

Topics (2/2)

8. Stack and Queue ADT, Lab5 - Tux
9. Recursion, Lab6 - Navi
10. Analysis of Algorithms
11. Sorting, Lab7 - Math
12. Hashing, Lab8 - Most Freq Substring, PE2 :O
13. This review lecture, Lab9 - Math Again (opt)

Reading week – time to restudy the entire thing

Final assessment on Thursday, 24 Nov 16, EV

Past Exam Questions

AY2014-15 Sem 2, Q13

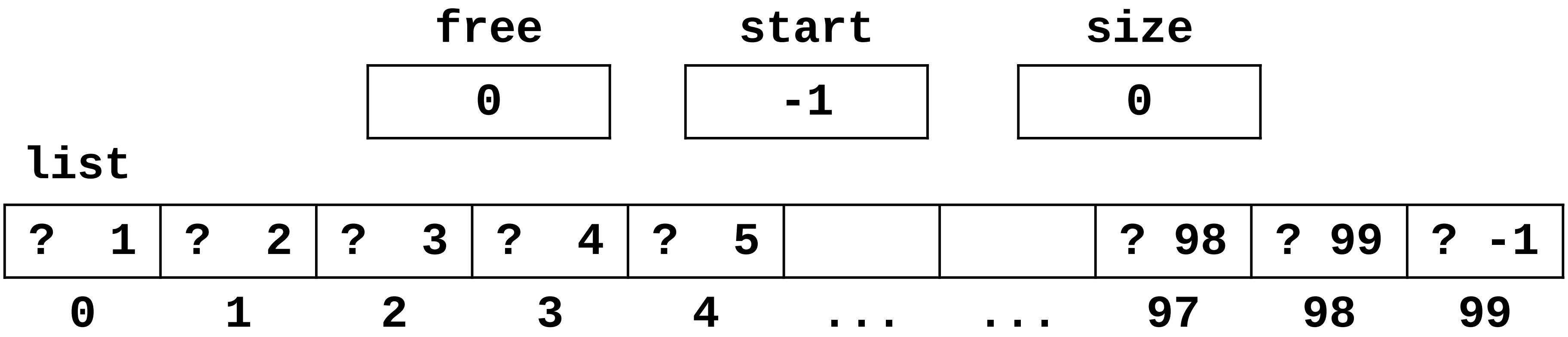
- A linked list can be simulated using array as follow:

```
struct listNode {
  char value; // value in the node
  int next;   // index of the next node in the list
};

int free;  // index of the first free slot
int start; // index of the first node in the list
int size;  // size of the list
listNode list[100]; // to store max 100 values
```

AY2014-15 Sem 2, Q13

- Initial state:



free	start	size
0	-1	0

list

0	1	2	3	4	...	...	97	98	99
? 1	? 2	? 3	? 4	? 5			? 98	? 99	? -1

- In a node, ? indicates unknown value, the number indicates the next slot in the list. Hence at the beginning, the whole list is a list of free nodes, and in the last node is –1 to indicate end of list and **`start`** is –1 to indicate empty list.

AY2014-15 Sem 2, Q13(B)

- Write C++ statements to **insert** a character ‘X’ to the **front** of the list. You should take care of the special cases and do proper housekeeping. If the list is full, throw an exception.

AY2014-15 Sem 2, Q13(C)

- Write C++ statements to **insert** a character ‘X’ to the **end** of the list. You should take care of the special cases and do proper housekeeping. If the list is full, throw an exception.

AY2014-15 Sem 2, Q13(D)

- Write C++ statements to **delete** the **first** node in the list. Again, you have to take care of the special cases and do proper housekeeping. If the list is empty, throw an exception.

AY2014-15 Sem 2, Q14

- Suppose you are given an array A[0 ... *n* – 1]* of **distinct sorted** integers that has been circularly shifted *k* positions to the right.
- For example, [45, 62, 1, 19, 22, 39] is a sorted array that has been circularly shifted *k* = 2 positions, while [22, 39, 45, 62, 1, 19] has been shifted *k* = 4 positions.

AY2014-15 Sem 2, Q14(A)

- Write a brute force algorithm to determine how many integers are shifted, that is, to find *k*. What is the complexity of your algorithm?

AY2014-15 Sem 2, Q14(B)

- Design an O(log *n*) time algorithm for finding the shift *k*.

AY2014-15 Sem 2, Q16(A)

- Describe the worst-case running time of the following functions in big-O notation in terms of the variable *n*.

```
void fun3(int n) {
  for (int i = 0; i < 1000; ++i) {
    for (int j = 0; j < n; ++j) {
      for (int k = 0; k < j; ++k)
        cout << "k = " << k;
      for (int m = 0; m < i; ++m)
        cout << "m = " << m;
    }
  }
}
```

AY2014-15 Sem 2, Q16(B)

- Describe the worst-case running time of the following functions in big-O notation in terms of the variable *n*.

```
void fun4(int n, int m) {
  if (n < 0) return;
  if (n < m)
    fun4(n/2, m+1);
  else
    fun4(n/2, m);
}
```

AY2014-15 Sem 2, Q16(C)

- Describe the worst-case running time of the following functions in big-O notation in terms of the variable *n*.

```
void fun5(int n, int x, int y) {
  for (int i = 0; i < n; ++i) {
    if (x < y)
      for (int j = 0; j < n*n; ++j)
        cout << "j = " << j;
    else
      cout << "i = " << i;
  }
}
```

AY2014-15 Sem 2, Q16(D)

- Describe the worst-case running time of the following functions in big-O notation in terms of the variable *n*.

```
void fun6(int n) {
  if (n > 2) {
    for (int i = 0; i < n; ++i) {
      int j = 0;
      while (j < n) {
        cout << "j = " << j;
        j++;
      }
    }
    fun6(n/2);
  }
}
```

AY2014-15 Sem 2, Q18(A)

- Write a recursive method to check whether a given integer is a prime number.

```
// Pre-condition: n is a positive integer greater than 1.
// Post-condition: return true if n is a prime number, false otherwise.
bool is_prime(int n) {

}

// You may have to write a helper method that is recursive.
```

AY2014-15 Sem 2, Q18(B)

- Write a recursive method to print all the prime numbers between 2 and *n* inclusive. You can call your method in Q18(A).

```
// Precondition: n is a positive integer.
// Postcondition: all the prime numbers between 1 and n
//                are printed in ascending order.
void print_prime(int n) {






}
```

Exam Matters

Final Exam (1/2)

- Date: Thursday, 24 November 2016
- Time: 5pm – 7pm
- Venue: To be announced by Registrar's Office

- Weightage: 40%
- Scope: Everything covered in lectures, tutorials and labs
  - ≈ 21%: material before recess week
  - ≈ 79%: material after recess week

Final Exam (2/2)

- Format
  - Section A, 8 “short” questions
    - can be answered with pseudocode, i.e. in English
  - Section B, 3 “medium” questions
    - need to be answered using C++ code, minor bugs are “OK” as long as we can understand your intention/algorithm
- Open book
  - You can bring anything
  - Calculators allowed, but not really needed
- Write answer in the boxed space provided
  - One free-to-use extra last page as in Midterm Test
  - Bug generally, if you write too long, your answer is likely wrong
  - You may use pencil to write your answers
- Read through all questions first before answering

Consultation Hours

- Email me or Ivan (full time TA) to make appointment during reading week
  - Steven stevenhalim@gmail.com
  - Ivan ictm@u.nus.edu
  - Please do not touch the other TAs
    - They are part-timer who have their own exams too

All the best for your Exams!