Q&A sessoin

28 Sept 10-12

Why the leak?

- You will see the exact format of the paper.
- Help you to save time in reading the paper.
- Help you to prepare mentally and maybe plan your strategy.
- To know the DOs and DONTs

Q5

- hasSpecial, shown below, returns true if there exist two unique numbers in the sequence that sum to a third number in the sequence. Assume n is the size of the sequence.

```
bool hasSpecial (vector<Integer> c) {
        for (int i = 0; i < n; ++i)
            for (int j = i + 1; j < n; ++j)
                  for (int k = 0; k < n; ++k)
                        if (c[i] +c[j] == c[k]) return true;
        return false;
}
```

Q5- cont

Let C1 be the running time of hasSpecial.

Let C2 be the running time of hasSpecial when a linkedlist is used instead of a vector. The list provides the function retrieve(i) where i is the index of the item to be retrieved. The if statement is replaced by if (retrieve[i] + retrieve[j] == retrieve[k]) return true;

What is the complexity C1 and C2?

a) C1 = $O(n^3)$, C2 = $O(n^3)$
b) C1 = $O(n^3)$, C2 = $O(n^4)$
c) C1 = $O(n^4)$, C2 = $O(n^3)$
d) C1 = $O(n^4)$, C2 = $O(n^4)$
e) None of the above.

equals, shown below, returns true if the two vectors have the same size and contain the same elements in the same order. Assume n is maximum of the sizes of the vectors.

```
bool equals (vector<Integer> lhs, vector<Integer> rhs) {
    if (lhs.size() != rhs.size()) return false;
    for (int i = 0; i < lhs.size(); ++i)
        if (lhs[i] != rhs[i]) return false;
    return true;
}
```

What are the time complexities of the best case and the worst case?

a) O(log n), $O(n^2)$
b) O(log n), O(n)
c) O(1), O(log n)
d) O(1), O(n)
e) None of the above

13. **[15 marks]**

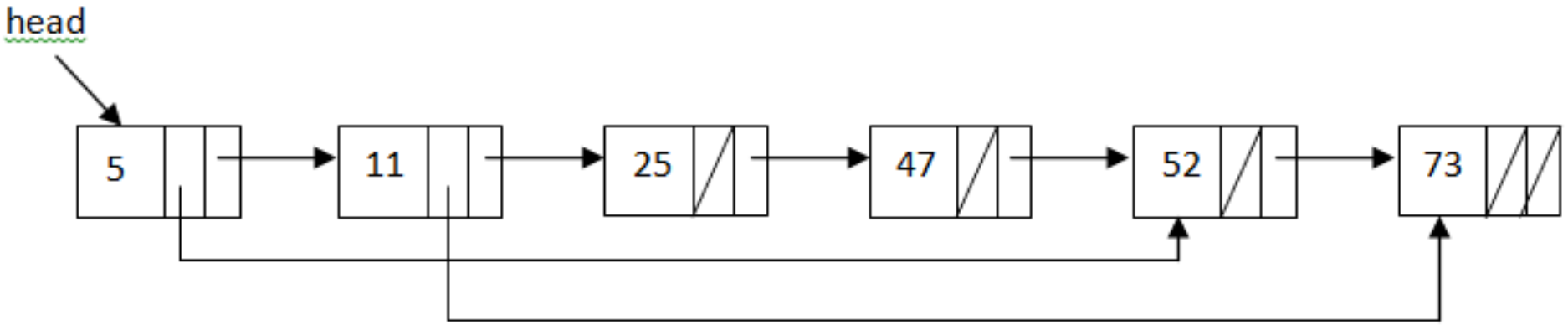
Assume the following KListNode declaration:

```
class KListNode {
        private int _item;      // value in the node
        private KListNode _kNext;   // points to a node K steps ...
                                    //    down the list
        private KListNode _next;     // the usual next pointer

        public int getItem() {return _item;}
        public KListNode getKNext() {return _kNext;}
        public KListNode getNext() {return _next;}

        // constructors and other accessors and mutators not listed
here
}
```

Each node in a LinkedList created from KListNode class has two pointers: one points to the next node and the other points to a node that is k steps down the list. In the following example, k = 4.



(a) Searching on a linked list is normally a O(*n*) operation as we need to perform linear search even if the values are sorted. With the _kNext pointer, the efficiency for searching on a sorted KLinkedList can be improved.

Write a static method to search for a value given a head pointer pointing to the first node of the list and an integer value. The method should return the reference to the node containing the value or return null if the value is not found in the list.

[9 marks]

13. (continued…)

(b) While the _kNext pointers improve the efficiency for searching on a sorted linked list, it would require more codes for inserting node to and removing node from the list. In particular, more nodes need to be modified when either operation is executed.

(i) What are the minimum and maximum number of nodes affected when a node is inserted into such a linked list? That is, how many _next pointers and _kNext pointers need to be changed? [3 marks]

(ii) What are the minimum and maximum number of nodes affected when a node is removed from such a linked list? That is, how many _next pointers and _kNext pointers need to be changed? [3 marks]

You have a set of data that represent students taking course CS1100. Each of the students has a *unique integer* student ID and is assigned a numeric grade between 0-100.

You are given the following four scenarios. Each scenario has an assumption and a set of operations to support. You can assume that the scenarios are independent of each other, and no other operations beside those given in the scenario needs to be supported.

For each of the scenarios (a) to (d), do the following:

- Describe an efficient data structure you can use to support the operations given.
- Describe how you support the operations.
- Give the running time of the operations using your data structure.

You can refer to any of the data structures you have learned in CS1102, combine the data structures, or invent your own. Keep your answer short but precise. Here is a sample answer:

"Store the students in a linked list. Insertion can be done in $O(1)$ time by always inserting the student at the head of the linked list. Retrieval can be done in $O(N)$ by searching through the linked list sequentially."

(a) **Assumption:** There are no more than 20 students. The student IDs are numbered from 1 to 20.

Operations:

- Insert a student.
- Given the ID of a student, retrieve his or her grade.

(b) **Assumption:** None

Operations:

- Insert a student.
- Retrieve the ID of a student with lowest grade.

(c) **Assumption:** Grades are integer numbers.

Operations:

- Insert a student.
- Given two integers X and Y between 0 and 100 ($X \leq Y$), retrieve the students whose grade are between X and Y.

(d) **Assumption:** Student IDs can be arbitrarily large integers.

Operations:

- Insert a student.
- Given two integers X and Y ($X \leq Y$), retrieve the students whose ID falls between X and Y.

Question 5 (10 marks)

Given an unsorted array of int, describe an **efficient** function to list all numbers in the array with values within the range [x..y]

(a) When the result is not required to be sorted

(b) When the result should be sorted.