

Data Structures (CS2001)

Midterm 1 Exam

Date: Mon Sep 23, 2024

Course Instructor(s)

ZA, SK, MN, SF, AK, MM, UN,

UH, AK, FA

Total Time (Hrs.): 1

Total Marks: 25

Total Questions: 3

Roll No

Section

15 + 2 + 5

Student Signature

Note: Solve Q2 and Q3 on the question Paper and attach it with answer sheet.

CLO # 1: Demonstrate basic concepts of data structure and algorithms

Q. No 1:

(Marks: 15)

Given a char singly linked list class **List**, write the function **DeleteLongestSequence** (char ch) in the **List** class. The above function takes a char ch as input, finds the first occurrence of the longest sequence of given ch in the List, and removes it from the given linked list. The function **DeleteLongestSequence** (char ch) should run in $O(N)$ time.

Examples:

Given ch = 'b' and list c->b->a->b->b->b->k->c->b->m->b->b->s

After executing function **DeleteLongestSequence** list becomes c->b->a->k->c->b->m->b->b->s

Given ch = 'b' and list = a->b->k->c->b->m->b->b->s->b->b->t

After executing function **DeleteLongestSequence** list becomes a->b->k->c->b->m->s->b->b->t

```
void main() {  
    List L1;  
    L1.inputList();  
    L1.DeleteLongestSequence (char ch)  
}
```

```
class List {  
    public:  
        ...  
        void inputList();  
        void DeleteLongestSequence (char ch)  
  
    private:  
        Node * head, * tail;  
};
```

The class **List** is not implemented with dummy (sentinel) nodes.

If you use any function (search or delete) of the **List** class, provide its code.

Note: Code should be efficient (space and time), well structured, and properly commented.

CLO # 2: Evaluate different data structures in terms of memory complexity and time

Q2: Give an estimate of $T(N)$. Solve here and give a $T(N)$ estimate for each line of code. (Marks: 5)

```
long operationCounter = 0;
void epsilon(int n) {
    for (int i = 0; i < n; i++)
        for (int j = 1; j < i; j *= 2)
            operationCounter++;
}
void zeta(int n) {
    for (int i = 0; i < n; i++)
        for (int j = 0; j < i; j=j+2)
            for (int k = j; k > 0; k /= 3)
                operationCounter++;
}
int main() {
    int n; cin >> n;
    operationCounter = 0;
    epsilon(n);
    zeta(n);
    return 0;
}
```

$T(N)$ for each line

$2n+2$
 $\frac{n+1}{2}$
 $\frac{n+1}{2}$
 $2n+2$
 $\frac{n}{2}(2n+2)$
 $\frac{n+1}{2}$
 2
 1
 $3n+3$
 $\frac{5n^2}{3} + 3n + \frac{10}{3}$

$3n+3$
 $\frac{5n^2}{3} + 3n + \frac{10}{3}$

$\frac{2n^2}{3} + \frac{4n}{3} + n^2 + n + 2n + 2$
 $= \frac{5n^2}{3} + 3n + \frac{10}{3}$

$T(N)$ estimate $= 1 + 2 + 1 + 3n + 3 + \frac{5n^2}{3} + 3n + \frac{10}{3}$

$$T(N) = \frac{5n^2}{3} + 6n + \frac{31}{3} = O(n^2)$$

~~Q2: 2n~~

CLO # 1: Demonstrate basic concepts of data structure and algorithms

Q3:

(Marks: 5)

A template-based Stack class is implemented. The pop operation pops and returns the element.

- Describe in a line what the Mystery function does.
- Show the content of Stack S before and after executing the Mystery function.

```
void Mystery(stack<int>& s, int k)
{
```

```
    Stack<int> s1, s2;
    while (!s.empty())
        s1.push(s.pop());
    // reverse stack into s1
    for (int i = 0; i < k; i++)
        s2.push(s1.pop());
    // for k, push from s1 to s2
    while (!s2.empty())
        s.push(s2.pop());
    // push them back to s (reverse)
    while (!s1.empty())
        s.push(s1.pop());
    // remaining back to s (orig)
}
```

```
int main() {
```

```
    Stack<int> S;
    for (int i=0; i<15; i++)
        S.push(i);
```

// → show stack contents and clearly mention the top

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
Base Top

```
    Mystery (S, 9);
```

// → show stack contents and clearly mention the top

8, 7, 6, 5, 4, 3, 2, 1, 0, 9, 10, 11, 12, 13, 14
Base Top

```
    return 0;
}
```

Ans. i) Mystery function reverses the order of first k elements from base (last k elements in stack).