

CSCI 2421: Data Structures and Program Design

Fall 2020 Midterm Exam

Name: Kerry Gip

Score: / 72

This is a take home test. Please make a note of the following:

- This is open book / open notes exam.
- You can also use a calculator.

Due Date: October 11, 2020 @ 11:55 PM – You must submit the test before this deadline. I am not going to make any exception to this rule.

GOOD LUCK!

- 1) Multiple Choice Questions: Please select the best answer from the given list for the following statements/questions. **(1 point each)**

A) What is the big O for the following:

$$f(n) = 5n^2 + \log n + 1$$

i) n^2

ii) 1

iii) $\log n$

iv) $5n^2 + \log n + 1$

B) Given two algorithms with the following running time: Algorithm A: $Ta(n) = 4n + 20$ and

Algorithm B: $Tb(n) = 2n^2 + 10$

Which of the following statements are correct?

i) Algorithm A is faster than algorithm B for $n > 100$.

ii) Algorithm B is faster than algorithm A for $n > 100$.

C) Recursion is a problem-solving process which breaks a problem into identical but smaller problems.

i) True

ii) False

D) A recursive function must have one or more cases that provide solution that does not require recursion.

i) True

ii) False

E) All recursive function can be converted into iterative function.

i) True

ii) False

F) A function template is not an actual function.

i) True

ii) False

G) A function template can operate with:

i) an integer

ii) any type of data

iii) a string

iv) a Rectangle object

H) A class template can be used to create a container class which can store integers only.

i) True

ii) False

I) An abstract data type or ADT is a collection of objects and operations.

i) True

ii) False

J) In C++ a linked list is implemented as a ____ STL.

i) vector

ii) array

iii) list

iv) queue

K) Which of the following characterizes a stack?

i) first in, first out

ii) first in, never out

iii) last-in, first-out

iv) last in, last out

L) Which of the following characterizes a queue?

i) first in, first out

ii) first in, never out

iii) last-in, first-out

iv) last in, never out

- 2) Determine the running time ($T(n)$) for the following loops. Just write down the expression. You don't have to simplify it. You must show the work to get credit. **(10 Points Each)**

A)

```
for (maxElement = n; maxElement > 0; maxElement--)
{
    for (index = 0; index < maxElement; index++)
    {
        System.out.println (index);
    }
}
```

- 2) Determine the running time ($T(n)$) for the following loops. Just write down the expression. You don't have to simplify it. You must show the work to get credit. **(10 Points Each)**

A)

```
for (maxElement = n; maxElement > 0; maxElement--)
{
    for (index = 0; index < maxElement; index++)
    {
        System.out.println (index);
    }
}
```

Inner loop iterates n times
Outer loop iterates n times
 $T(n) = N^2$

Outer loop runs at n times and
inner loop will execute $n, n-1, n-2$ for decreasing values of maxElement
 $T(n) = n(n+1)/2 = (n^2 + n)/2$. n^2 is the largest so
 $T(n) = O(n^2)$

B)

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        for (k = 0; k < n; k++)
        {
            System.out.println (i, j, k);
        }
    }
}
```

B)

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        for (k = 0; k < n; k++)
        {
            System.out.println (i, j, k);
        }
    }
}
```

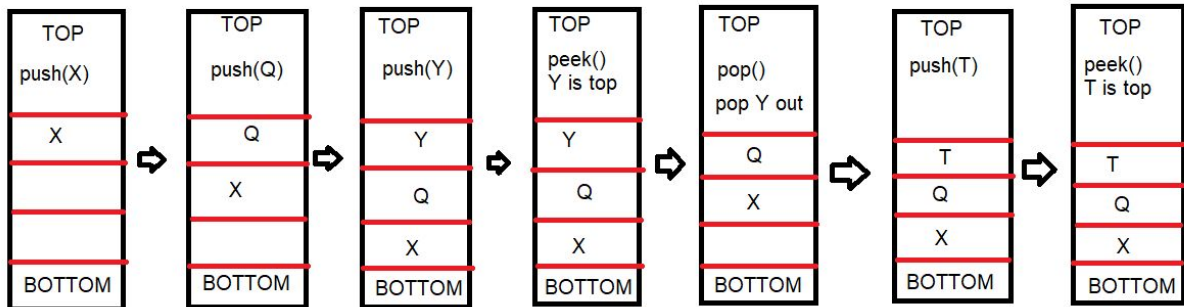
Outer loop iterates n times
Middle loop iterates n times
Inner loop iterates n times
 $T(N) = N*N*N = N^3$

Three loops all running at n time. $T(n) = n*n*n$ so
 $T(n) = O(n^3)$

- 3) Draw the resultant stack after each of the following operations. Label the top entry of your stack. **(10 points)**

push(X), push(Q), push(Y), peek(), pop(), push(T), peek()

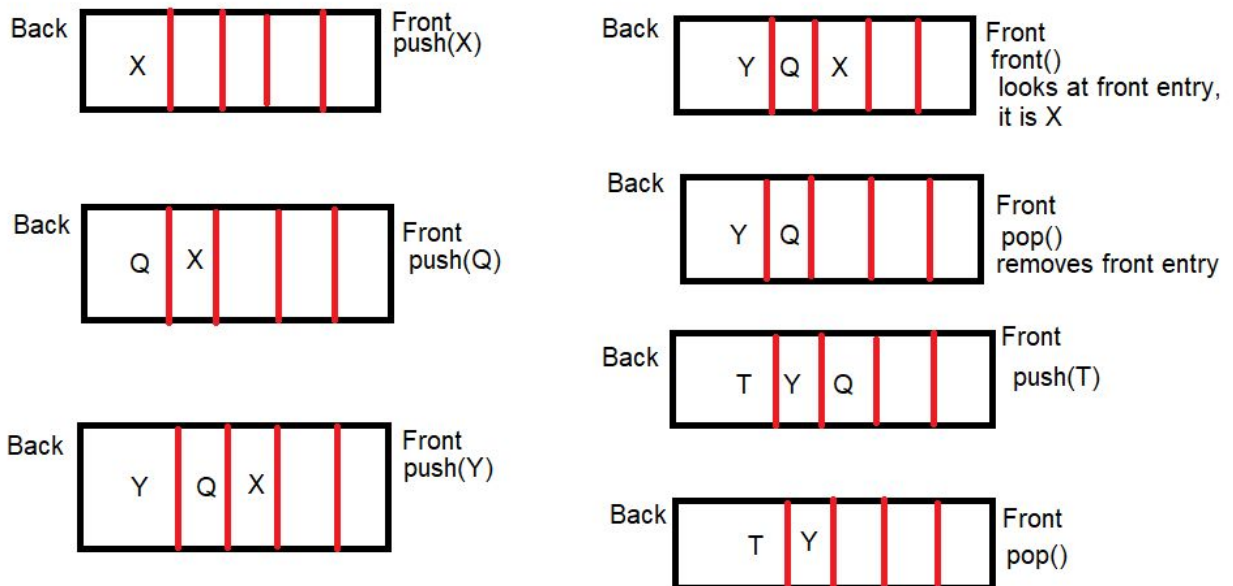
push(X), push(Q), push(Y), peek(), pop(), push(T), peek()



- 4) Draw the resultant queue after each of the following operations. Label the front and back entry of your queue. **(10 points)**

push(X), push(Q), push(Y), front(), pop(), push(T), pop()

push(X), push(Q), push(Y), front(), pop(), push(T), pop()



- 5) Given a linked list implemented with a C++ list STL. Write a function template to find an item in the list. You can assume that `cmp` is the Comparator object which contains the `isEqual` function. `isEqual` function will take two objects and return a true if those two objects are equal. You can use the `Iterator` to iterate through the linked list. **(10 points)**

```

1  #include<iostream>
2  #include<list>
3  #include<iterator>
4  using namespace std;
5
6  template <typename Comparable>
7  const Comparable & isEqual(const list<Comparable> & same) {
8  list<Comparable>::iterator i;
9
10 for (i = list.begin(); i != list.end(); i++){
11     if Comparable.isEqual(*i, same)
12         return true;
13     else
14         return false;
15 }
16 }

```

- 6) Given two integers, start and end, where end is greater than start, write a recursive C++ function that returns the sum of the integers from start through end, inclusive. Example: If start is 5 and end is 10 then the function will return: 45 which is sum of 5, 6, 7, 8, 9, and 10. **(10 points)**

```

int sum (int start, int end)
{

```

```

//Given two integers, start and end, where end is greater than start,
//write a recursive C++ function that returns the sum of the integers from start through end, inclusive.
//Example: If start is 5 and end is 10 then the function will return: 45 which is sum of 5, 6, 7, 8, 9, and 10. (10 points)

#include<iostream>
using namespace std;

int sum (int start, int end)
{
    if (start == end){
        return end; // if numbers are the same
    }
    else
        return sum(start+1, end) + start; // if not same
}

int main(){
    int start, end;
    int total = 0;
    cout<<"Starting integer is: " << endl;
    cin>> start;
    cout<<"Ending integer is: "<< endl;
    cin >> end;

    total = sum(start, end);
    cout<< "Total is: " << total << endl;
}

```