CPE 360 Midterm I
Prof. Iyengar
3/30/21
Mark Tuhacek

1. The benefits of creating data structures that use heap over continuous memory locations are that heap is more dynamic, and there is larger memory available and allows for dynamic resizing(grow or shrink). Instead of using the worst-case scenario memory allocation in the continuous memory, heap allows for dynamic allocation and reallocation as needed.

An example of this is an online shopping cart, where we cannot estimate how much the shopper will buy. The cart needs to be dynamic so that the shopper can add, remove, and update as needed. Allocating space in the heap fits this scenario much better than pre-allocating continuous memory.

2.

(a) 3n^2 + 4n^3 + 5 = O(n^2)
false, would be of the order of O(n^3) since n^3 > n^2

(b) 8logn + 4n = O(n)
This is true, n has precedence over logn, so the order is O(n)

(c) 2^n + 4n^3 = O(n^3)
false, 2^n is dominant over n^3, so the order would be O(2^n)

(d) 2n^2 + 3n = O(n^2)
True, n^2 > 3n

(e) 2^n +n^2 +O(n) = O(n^2)
false, again 2^n > n^2, so order would be O(2^n)

3. Fibonacci recursive function:

```
#include <iostream>
using namespace std;
int getNthTermFibonacci(int n) {
   static int loopCount=0;
   int sum=0;

   loopCount++;
  if ((n == 1) || (n == 0)) {
   loopCount--;
   if (loopCount == 0)
      cout << n << " ";
   return (n);
  } else {
```

```
    sum = (getNthTermFibonacci(n - 1) + getNthTermFibonacci(n - 2));
    loopCount--;
    if (loopCount == 0)
      cout << sum << " ";
  return sum;
 }
}
int main() {
 int n, i = 0;
 cout << "Enter the Fibbonaci term : ";
 cin >> n;
 while (i < n) {
  getNthTermFibonacci(i++);
 }
 return 0;
}
```

4. [10 points] Why are linked-lists better than arrays? List at least three advantages to a linked-list structure over static arrays?

With linked-lists, you can insert at a position such as between two chunks, at the front, middle, or end, as well as delete at a specific position. The size of the linked-list can also be changed as needed, as opposed to the fixed size of a static array.

5. [5 points] What is the difference between a “const” variable and a “static” variable? When do you use each of these in your program?

A “const” variable is a variable that is set to a specific value at compile time and cannot be changed at run time. A static variable can change during run time but is only initialized once and it keeps that value across multiple executions of the function.

An example of using a constant variable is calculating the area of a circle, using the variable x = 3.14 for pi. An example of using a static variable is using a property named schoolName in a class that represents a class offering. The schoolName is initialized once and is the same for all instances of that class. Another example of using a static variable is using it as a function loop counter to count how many times a function has been called.

6. [5 points] What is the “signature” of a function?

A function signature specifies the type of data being returned, the function name, and the data types, and the number of input parameters.

```
void insertAtPosition(int listInput, int listPosition)
```