University of Nevada, Las Vegas Computer Science 477/677 Fall 2022

Assignment 2: Due Friday September 16, 2022, Midnight

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. Please follow Mr. Want's instructions on how to submit your completed assignment.

1. Solve each recurrence, writing O, Ω, or Θ, whichever is most appropriate.

   (a) $F(n) = 4F\left(\frac{n}{2}\right) + 5n^2$

   (b) $f(n) = f(n-1) + n.$

   (c) $f(n) = f\left(\frac{n}{2}\right) + f\left(\frac{n}{3}\right) + n$

   (d) $f(n) = f(\sqrt{n}) + 1.$

   (e) $f(n) = 2f(\sqrt{n}) + \log n$

   (f) $H(n) \leq 2H\left(\frac{n}{2}\right) + n$

   (g) $g(n) = 2g(n-1) + 1$

   (h) $G(n) \geq G(n-1) + \lg n$

   (i) $H(n) \leq 2H(\sqrt{n}) + 4.$

(j) $K(n) = K(n - 2\sqrt{n} + 1) + n.$

(k) $F(n) \leq F\left(\frac{n}{5}\right) + F\left(\frac{7n}{10}\right) + n$

(l) $F(n) = 2F\left(\frac{2n}{3}\right) + F\left(\frac{n}{3}\right) + n$

(m) $f(n) = 1 + f(\log n)$

2. Find the asymptotic time complexity, in terms of n, for each C++ code fragment. Assume $n \geq 0$. These problems are similar to the ones above, except that you have to read the code, write the recurrence, then solve the recurrence.

   (a)
```
void f(int i)
 {
  for(int j = 0; j < i; j++)
   cout << "hello world" << endl;
  if(i > 0) f(i/2);
  if(i > 0) f(i/2);
 }
int main()
 {
  f(n);
  return 1;
 }
```

(b)
```
void f(int i)
 {
  if(i > 0)
   {
    for(int j = 0; j < i*i; j++);
     cout << "hello world" << endl;
    f(2*i/3);
    f(i/3);
    f(2*i/3);
   }
 }
int main()
 {
  f(n);
  return 1;
 }
```

(c) You have nine coins, one of which is counterfeit. The eight good coins all weigh the same, but the counterfeit coin is slightly lighter. You have a balance scale. How can you find the counterfeit coin with at most two weighings?

3. The Coin-row problem: there is a row of n coins whose values are positive integers $C_0, C_1, C_2, \ldots, C_{n-1}$, not necessarily distinct. The goal is to pick up the maximum amount of money subject to the constraint that no two coins adjacent in the initial row can be picked up.

   Pick **one** of the three paradigms listed below, and write an algorithm for the coin-row problem using that paradigm.

   1. Greedy
   2. Dynamic Programming
   3. Recursion

The loop invariant of the following code is that sum is $A[0] + A[1] + \cdots A[i]$.

```
int sum = A[0];
int i = 0;
while(i < n)
 {
  i = i+1;
  sum = sum+A[i];
 }
cout << "The sum is " << sum << endl;
```

4. What is the loop invariant of the following code? What is the output?

```
int m = A[0];
int i = 0;
while(i < n)
 {
  i = i+1;
  if(A[i] > m) m = A[i];
 }
cout << m << endl;
```

5. What is the loop invariant of the following code? What is the output?

```
float x;
cin >> x;
int k;
cin >> k;
assert(k >= 0);
int i = 0;
float y = 1;
while(i < k)
  {
   i = i+1;
   y = y*x;
  }
cout << y << endl;
```