University of Nevada, Las Vegas Computer Science 477/677 Fall 2020

Assignment 5: Due Thursday October 1, 2020

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. Your answers must be written in a pdf file and emailed to the graudate assistant, Tandreana Chua `chuat4@unlv.nevada.edu` , by midnight October 1. Your file must not exceed 5 megabytes, and must print out to at most 4 pages.

All answers to the first two questions will be found in the following list of functions: $\sqrt{n}$, $n\sqrt{n}$, n, n^2, n^3, n^4, $\log n$, $\log\log n$, $\log^2 n$, $n\log n$, $n^2\log n$.

1. Give an asymptotoic time complexity to each of these code fragments. The answer should be expressed using Θ notation, except for (1j).

   Instead of just guessing, or asking for the answer from someone, first make a serious effort to solve them yourself using the techniques I've shown you. In some cases, it might help to actually implement the code. Here is a suggestion on how you could do that.

```
cout << "Enter a positive integer: ";
cin >> n;
cout << endl;
int kount = 0;
for(int i = 0; i < n; i++)
  {
   kount++;
  }
cout << "for n = " << n << " kount = " << kount << endl;
```

(a)
```
for(int i = 0; i < n; i++)
    for(int j = 0; j < n; j++)
```

(b)
```
for(int i = 1; i < n; i = 2*i)
```

(c)
```
for(int i = 1; i < n; i++)
    for(int j = n; j > 0; j = j/2)
```

(d)
```
for(int i = 0; i < n; i++)
    for(j = i; j > 1; j = j/2)
```

(e)
```
for(int i = 0; i < n; i++)
    for(j = n; j > i; j = j/2)
```

(f)
```
for(int i = n; i > 0; i = i/2)
    for(int j = i; j > 0; j = j/2)
```

(g)
```
for(int i = 0; i < n; i++)
    for(int j = 0; j < i*i; j++)
      if(j%n == 0)
        for(int k = 0; k < j; k++)
```

(h) `for(int i = 2; i < n; i = i*i)`

(i)
```
for(int i = 0; i < n; i++)
  for(int j = 2; j*j < i; j++)
```

(j) The next problem cannot be given an answer using Θ. There are two answers you must give; an upper bound using O notation, and a lower bound using Ω notation. The two functions are different.

```
for(int i = 2; i < n; i = i*i)
  for(int j = 0; j < i; j++)
```

2. For each of these recursive functions, let T(n) be the time complexity of the function with input n. In each case, write a recurrence for T(n), and then solve that recurrence.

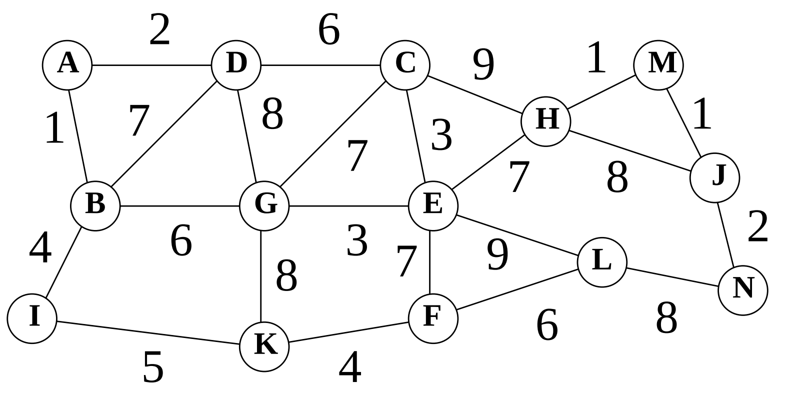
(k)
```
void f(int n)
 {
  if(n > 0)
   {
    for(int i = 1; i < n; i++)
     for(int j = 1; j < n; j++)
      cout << "hello!" << endl
    f(n/2); f(n/2);
   }
 }
```

(l)
```
void g(int n)
 {
  if(n > 0)
   {
    for(int i = 1; i < n; i++)
     for(int j = 1; j < n; j++)
      cout << "hello!" << endl
    g(n/2); g(n/2); g(n/2); g(n/2);
   }
 }
```

(m)
```
void h(int n)
 {
  if(n > 0)
   {
    for(int i = 1; i < n; i++)
     for(int j = 1; j < n; j++)
      cout << "hello!" << endl
    h(n/2); h(n/2); h(n/2); h(n/2); h(n/2); h(n/2); h(n/2); h(n/2);
   }
 }
```

3. Walk through heapsort, using the method shown in class, for the array: **A Q R B X S M L N T**

4. Walk through Kruskal's algorithm for a minimum spanning tree of the following weighted graph, showing the steps of union/find, and using path compression.


A
D
C
M
H
J
B
G
E
L
N
I
K
F
2
6
9
1
1
7
8
3
7
7
8
2
4
6
3
7
9
8
6
8
5
4