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

Answers to Assignment 2: Due Monday February 7, 2022

Read the handouts recurrence.pdf, linLog.pdf, and bigOThOm.pdf.

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

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

   $F(n) = \Theta(n^2 \log n)$

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

   $f(n) = \Theta(n^2)$

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

   $f(n) = \Theta(n)$

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

   $f(n) = \Theta(\log \log n)$

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

   $f(n) = \Theta(\log n \log \log n)$

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

   $H(n) = \Theta(n \log n).$

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

   $g(n) = \Theta(2^n)$

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

   $G(n) = \Theta(n \log n)$

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

   $H(n) = O(\log n)$

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

   $K(n) = \Theta(n^{\frac{3}{2}})$

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

   $F(n) = O(n)$

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

   $F(n) = \Theta(n^2)$

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

   $f(n) = \Theta(\log^* n)$

2. Write the asymptotic time complexity for each code fragment, giving the answer in terms of n, using O, Ω, or Θ, whichever is appropriate.

(a)
```
for (int i=1; i < n; i++)
  for (int j=i; j > 0; j--)
    cout << "hello world" << endl;
```

$\Theta(n^2)$

(b)
```
for (int i=1; i < n; i=2*i)
  for (int j=i; j < n; j++)
    cout << "hello world" << endl;
```

$\Theta(n \log n)$

(c)
```
for (int i=1; i < n; i = 2*i)
  for (int j=1; j < i; j++)
    cout << "hello world" << endl;
```

$\Theta(n)$

(d)
```
for (int i=1; i < n; i++)
  for (int j=1; j < i; j = j*2)
    cout << "hello world" << endl;
```

$\Theta(n \log n)$

(e)
```
for (int i=1; i < n; i++)
  for (int j=i; j < n; j = j*2)
    cout << "hello world" << endl;
```

$\Theta(n)$

(f)
```
for (int i=2; i < n; i = i*i)
  cout << "hello world" << endl;
```

$\Theta(\log \log n)$

(g)
```
for (int i=1; i*i < n; i++)
  cout << "hello world" << endl;
```

$\Theta(\sqrt{n})$

(h)
```
for (int i=n; i > 1; i = i/2)
  for (int j=1; j < i; j=2*j)
    cout << "hello world" << endl;
```

$\Theta(\log^2 n)$

(i) This probem differs from the others. There is no Θ solution. However, there is an O solution, and there is an Ω solution. Find both.

```
for(int i = 2; i < n*n; i = i*i)
  for(int j = 0; j < i; j++)
    cout << "hello world" << endl;
```

$O(n^2)$, $\Omega(n)$

3. Find the asymptotic time complexity, in terms of n, for each C++ code fragment. Assume $n \geq 0$.

(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;
 }
```
$\Theta(n \log n)$

(b)
```
void f(int i,int j)
 {
  cout << "hello world" << endl;
  if(j > 0)
   f(i,j-1);
  else if(i > 0)
   f(i-1,i);
 }
int main()
 {
  f(n,n);
  return 1;
 }
```
$\Theta(n^2)$

(c)
```
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;
 }
```
$\Theta(n^2 \log n)$