

## Lecture Notes

1. Evaluate. Hint: all answers are integers. (Almost all)

- (i)  $\log_2 4 = 2$ , because  $2^2 = 4$ .
- (ii)  $\log_4 2 = \frac{1}{2}$ , because  $2 = \sqrt{4} = 4^{\frac{1}{2}}$ .
- (iii)  $\log_A A^2 = 2$  where  $A$  is any positive number larger than 1.
- (iv)  $x = 2$ , where  $\log_x 8 = 3$ .  $x = 2$ , since  $2^3 = 8$ .
- (v)  $x = 2$  where  $3 \log_x^2 9 = 2$ .  $x = 2$  because  $3^2 = 9$ .
- (vi)  $\log_4 0.0625 = -2$ , because  $4^{-2} = \frac{1}{4^2} = \frac{1}{16} = 0.0625$
- (vii)  $x = 1$ , where  $\log_7 x = 0$ .  $x = 1$ , because the logarithm of 1 is zero, regardless of the base.
- (viii)  $2^{\log_4 25} = \dots$ . Very tricky. Be in class.
- (ix)  $9^{\log_3 \sqrt{5}} = \dots$ . Very tricky. Be in class.

2. Give the asymptotic complexity, in terms of  $n$ , of each of the following code fragments. I have rewritten the problems in (more or less) order of difficulty.

- (i) 

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

\vskip 0.1in  
 $\Theta(n)$
- (ii) 

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

\vskip 0.1in  
 $\Theta(n)$
- (iii) 

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

\vskip 0.1in  
 $\Theta(\log n)$
- (iv) 

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

\vskip 0.1in  
This code runs forever, since  $2 * 0 = 0$ .
- (v) 

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

\vskip 0.1in  
 $\Theta(n^2)$
- (vi) 

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

\vskip 0.1in  
 $\Theta(\sqrt{n})$

```
(vii)  for(int i = 0; i < n; i++)
        for(int j = 1; j < n; j = 2*j)
            cout << "hello world" << endl;
```

$\Theta(n \log n)$

```
(viii) for(int i = 0; i < n; i = i+1);
        for(int j = 0; j < i; j++);
            cout << "Hello world!" << endl;
```

$\Theta(n^2)$

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

$\Theta(n)$

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

$\Theta(n)$

```
(xi)   for(int i = 0; i < n; i++)
        for(int j = i; j > 1; j = j/2)
            cout << "hello world" << endl;
```

$\Theta(n \log n)$

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

```
(xiii) for(int i = 0; i < n; i = i+1);
        for(int j = i; j < n; j++);
            cout << "Hello world!" << endl;
```

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

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

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

```
(xvii) for(int i = 1; i < n; i++)
```

```

        for(int j = i; j > 0; j = j/2)
            cout << "hello world" << endl;

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

(xix)    for(int i = 1; i < n; i++)
            for(int j = i; j < n; j=j+j)
                cout << "hello world" << endl;

(xx)     for(int i = 1; i < n; i++)
            for(int j = n; j > i; j = j/2)
                cout << "hello world" << endl;

(xxi)    for(int i = 1; i < n; i = i/2);
            cout << "Hello world!" << endl;

(xxii)   for(int i = 1; i < n; i = i+i);
            cout << "Hello world!" << endl;

(xxiii)  for(int i = 1; i < n; i=i+i;
            for(int j = 0; j < i; j++)
                cout << "hello world" << endl;

(xxiv)   for(int i = 1; i < n; i=i+i;
            for(int j = i; j < n; j++)
                cout << "hello world" << endl;

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

(xxvi)   for(int i = 2; i < n; i++)
            for(int j = 1; j < log i; j++)
                cout << "hello world" << endl;

(xxvii)  for(int i = 2; i < n; i=i*i)
            cout << "Hello world!" << endl;

(xxviii) for(int i = n; i > 1; i=i/2;
            for(int j = 0; j < i; j++)
                cout << "hello world" << endl;

(xxix)   for(int i = n; i > 1; i=i/2;
            for(int j = i; j < n; j++)
                cout << "hello world" << endl;

(xxx)    for(int i = 2; i < n; i = i*i)
            cout << "Hello world!" << endl;

```

```

(xxxxi)   for(int i = n; i > 2; i = sqrt(i))
           cout << "Hello world!" << endl;

(xxxii)   for(int i = 1; i < n; i = i++)
           for(int j = 2; j < n; j=j*j)
           cout << "Hello world!" << endl;

(xxxiii)  for(int i = 1; i < n; i = i++)
           for(int j = 2; j < i; j=j*j)
           cout << "Hello world!" << endl;

(xxxiv)   for(int i = 2; i < n; i = i++)
           for(int j = i; j < n; j=j*j)
           cout << "Hello world!" << endl;

```

3. Find asymptotic complexities of values computed by code, each in terms of  $n$ .

(i) The asymptotic complexity of `mystery1(n)` is  $\Theta(n \log n)$ .

```

float mystery1(int n)
{
    float totl = 0;
    for(int i = 1; i < n; i++)
        totl += log(i);
    return totl;
}

```

Use integration. The answer is  $\sum_{i=1}^n \log(i)$  which is approximated by the integral

$$\int_{x=1}^n \ln x dx = (x \ln x - x) \Big|_1^n = (n \ln n - n) - (1 \ln 1 - 1) = \Theta(n \log n).$$

(ii) The asymptotic complexity of `mystery2(n)` is  $\Theta(\log n)$ .

```

float mystery2(int n)
{
    if(n <= 1) return 1;
    else return 1 + mystery2(n/2);
}

```

4. Explain how to compute the  $n^{\text{th}}$  Fibonacci number  $F(n)$  in  $O(\log n)$  time using the recurrence

$$F(1) = F(2) = 1$$

$$F(n) = F\left(\frac{n-1}{2}\right) * F\left(\frac{n}{2}\right) + F\left(\frac{n+1}{2}\right) * F\left(\frac{n+2}{2}\right) \text{ for } n > 2$$

where, for any integer  $m$ , the expression  $m/2$  is always rounded down to an integer, as in C++. Be sure to state the asymptotic space complexity of your algorithm.