2) $s(n) = \sum_{i=1}^{n} \log i$

$$\sum_{i=1}^{n} \log i = \log 1 + \log 2 + \dots \log n$$
$$\le \log n + \log n + \log n + \dots + \log n$$
$$= n \log n$$

Therefore $\sum_{i=1}^{n} \log i \in O(n \log n)$

~~For~~

$$\sum_{i=1}^{n} \log i = \log 1 + \log 2 + \log 3 + \dots + \log n$$
$$\ge \log(n/2) + \log(n/2) + \log(n/2 + 1) + \log n$$
$$\ge \log n/2 + \log n/2 + \dots + \log n/2$$
$$= n/2 \log n/2$$

$\therefore \sum_{i=1}^{n} \log i \in \Omega(n \log n)$

3a)

```
int sum=0;
for(int n=N ; n>0 ; n/=2) } n, n/2 ..., n/4, iterations
  for(int i=0 ; i<n; i++)  >>   0,1,2 ... n "
    sum ++
```

$\therefore$ if the first for loop runs for $n \dots n/2$
the second loop runs for $i=0 \dots n-1$

$\therefore$ it runs for $\Theta(n/2) = \Theta(n)$

b) int sum=0

```
for (int i=1 ; i<N ; i*= 2)
    for (int j=0 ; j<i ; j++)
        sum++;
```

$i=1, 2i, 4i, \ldots\ldots i^2$

$i = 1, 2, 4, 8 \ldots\ldots$ O(N)

O(i)

$O(N) \times O(1) = O(N)$

int sum = 0

```
for (int i =1 ; i<N   i*=2)
    for (int j=0 ; j<n ; j++)
        sum++;
```

0, 1 . . . . n

i at m^{th} iteration $i = 2^{m-1}$

$2^{i-1} < i < 2^i$ so ∴ the outer loop runs exponentially at logn.

so $\log n \times n = O(n \log n)$

4 Prove by induction

$$\sum_{i=1}^{n} (2i-1) = n^2 \text{ for all } n \geq 1$$

* Basis: when $n=1$ $2(1)-1 = 1$, and $n^2 = 1$ so the identity holds.

* Induction hypothesis:
Suppose $\sum_{i=0}^{n} (2i-1) = n^2$ for some $n \geq 1$ for some $n \geq 1$

* Induction step: consider $n+1$

$$\sum_{i=1}^{n+1} 2i-1 = (2\cdot 1+1) + (2\cdot 2+1) + \dots + (2\cdot n-1) + (2\cdot(n+1)-1)$$

$$= \left[\sum_{i=1}^{n} 2i-1\right] + (2(n+1)-1)$$

$$= n^2 + (2(n+1)-1)$$ from the induction hypothesis

$$= n^2 + 2n + 1$$

$$= (n+1)^2$$

Therefore the identity holds for $n+1$ and by induction for all $n \geq 1$

1 $5, n^{0.01}, 4^{\log n}, (\log n)^5, 2n\log\log n, 5n, n^5, 5^n, n!, 2^{2n}$

5

```
Algorithm Missing(int[] P)
  Arrays.Sort(P)
    if P[0] is not equal to 0
        "The missing number is 0"
    End if
    ~~else~~ for (i=0 ; i< P.length-1 ; i++)
            j <- i+1
            diff <- P[j] - P[i]
            if diff != 1
                for x=1 ; x < diff ; x++
                    Print("the missing number is" + (P[i]+x))
                End for
            End if
    End for
```