

# Exercise

Express functions in A in asymptotic notation using functions in B.

A

$$5n^2 + 100n$$

$$A \in \Theta(n^2), n^2 \in \Theta(B) \Rightarrow A \in \Theta(B)$$

$$\log_3(n^2)$$

$$\log_b a = \log_c a / \log_c b; A = 2 \lg n / \lg 3, B = 3 \lg n, A/B = 2/(3 \lg 3)$$

B

$$3n^2 + 2$$

$$A \in \Theta(B)$$

$$\log_2(n^3)$$

$$A \in \Theta(B)$$

# Summations – Review

# Review on Summations

- **Linear Series (Arithmetic Series):** For  $n \geq 0$ ,

$$\sum_{i=1}^n i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

- **Quadratic Series:** For  $n \geq 0$ ,

$$\sum_{i=1}^n i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

# Review on Summations

- **Cubic Series:** For  $n \geq 0$ ,

$$\sum_{i=1}^n i^3 = 1^3 + 2^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{4}$$

- **Geometric Series:** For real  $x \neq 1$ ,

$$\sum_{k=0}^n x^k = 1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}$$

For  $|x| < 1$ , 
$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

# Review on Summations

- **Linear-Geometric Series:** For  $n \geq 0$ , real  $c \neq 1$ ,

$$\sum_{i=1}^n ic^i = c + 2c^2 + \cdots + nc^n = \frac{-(n+1)c^{n+1} + nc^{n+2} + c}{(c-1)^2}$$

- **Harmonic Series:**  $n$ th harmonic number,  $n \in \mathbb{I}^+$ ,

$$\begin{aligned} H_n &= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \\ &= \sum_{k=1}^n \frac{1}{k} = \ln(n) + O(1) \end{aligned}$$

# Review on Summations

- **Approximation by integrals:**

- For monotonically increasing  $f(n)$

$$\int_{m-1}^n f(x)dx \leq \sum_{k=m}^n f(k) \leq \int_m^{n+1} f(x)dx$$

- For monotonically decreasing  $f(n)$

$$\int_m^{n+1} f(x)dx \leq \sum_{k=m}^n f(k) \leq \int_{m-1}^n f(x)dx$$

# Search

- Given a array [2,7, 11,15], find the element with value 11.

```
Bool find (int a[], int n, int t){  
    for (int i = 0; i<n; i++)  
        if (a[i] == t)  
            return true;  
    return false;  
}
```

# Search

- Given an array [2,7, 11,15], find the element with value 11.

```
Bool find (int a[], int n, int t){  
    for (int i = 0; i < n; i++)  
        if (a[i] == t)  
            return true;  
    return false;  
}
```

Worse case	Best Case	Average Case
$O(n)$	$O(1)$	$O(n)$
$\Omega(n)$	$\Omega(1)$	$\Omega(n)$
$\Theta(n)$	$\Theta(1)$	$\Theta(n)$



# Search

- Binary search

```
Bool find(int a[], int n, int t){  
    int l = 0;  
    int r = n-1;  
    int m;  
  
    While (l <= r){  
        m = l + (r-l)/2;  
        if (a[m] == t) return true;  
        if (a[m] < t) l = m + 1;  
        else r=m-1;  
    }  
    Return false;}  
}
```

# Search

- Binary search

```
Bool find(int a[], int n, int t){
int l = 0;
int r = n-1;
int m;

While (l <= r){
    m = l + (r-l)/2;
    if (a[m] == t) return true;
    if (a[m] < t) l = m + 1;
    else r=m-1;
}
Return false;}
```

Worse case	Best Case	Average Case
$O(\log n)$	$O(1)$	$O(\log n)$
$\Omega(\log n)$	$\Omega(1)$	$\Omega(\log n)$
$\Theta(\log n)$	$\Theta(1)$	$\Theta(\log n)$

# Search

- Pros and Cons:
- Linear Search: unsorted / sorted list
- Binary Search: sorted list

# Sorting

- Insertion Sort
- Merge Sort
- Quick Sort

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	2.78	7.42	0.56	1.12	1.17	0.32	6.21	4.42	3.14	7.71

Iteration 0: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.


Array index	0	1	2	3	4	5	6	7	8	9
Value	2.78	7.42	0.56	1.12	1.17	0.32	6.21	4.42	3.14	7.71

Iteration 1: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	2.78	0.56	7.42	1.12	1.17	0.32	6.21	4.42	3.14	7.71




Iteration 2: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	2.78	7.42	1.12	1.17	0.32	6.21	4.42	3.14	7.71



Iteration 2: step 1.



# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

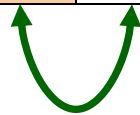
Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	2.78	7.42	1.12	1.17	0.32	6.21	4.42	3.14	7.71

Iteration 2: step 2.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	2.78	1.12	7.42	1.17	0.32	6.21	4.42	3.14	7.71

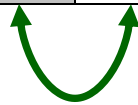


Iteration 3: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	2.78	7.42	1.17	0.32	6.21	4.42	3.14	7.71



Iteration 3: step 1.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.


Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	2.78	7.42	1.17	0.32	6.21	4.42	3.14	7.71

Iteration 3: step 2.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	2.78	1.17	7.42	0.32	6.21	4.42	3.14	7.71

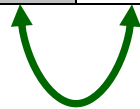


Iteration 4: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	1.17	2.78	7.42	0.32	6.21	4.42	3.14	7.71



Iteration 4: step 1.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.


Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	1.17	2.78	7.42	0.32	6.21	4.42	3.14	7.71

Iteration 4: step 2.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	1.17	2.78	0.32	7.42	6.21	4.42	3.14	7.71




Iteration 5: step 0.



# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	1.17	0.32	2.78	7.42	6.21	4.42	3.14	7.71



Iteration 5: step 1.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	1.12	0.32	1.17	2.78	7.42	6.21	4.42	3.14	7.71




Iteration 5: step 2.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.56	0.32	1.12	1.17	2.78	7.42	6.21	4.42	3.14	7.71

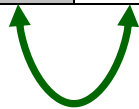


Iteration 5: step 3.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	7.42	6.21	4.42	3.14	7.71



Iteration 5: step 4.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.


Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	7.42	6.21	4.42	3.14	7.71

Iteration 5: step 5.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	6.21	7.42	4.42	3.14	7.71



Iteration 6: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.


Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	6.21	7.42	4.42	3.14	7.71

Iteration 6: step 1.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	6.21	4.42	7.42	3.14	7.71




Iteration 7: step 0.



# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	4.42	6.21	7.42	3.14	7.71



Iteration 7: step 1.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.


Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	4.42	6.21	7.42	3.14	7.71

Iteration 7: step 2.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	4.42	6.21	3.14	7.42	7.71




Iteration 8: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	4.42	3.14	6.21	7.42	7.71




Iteration 8: step 1.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	3.14	4.42	6.21	7.42	7.71



Iteration 8: step 2.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	3.14	4.42	6.21	7.42	7.71

Iteration 8: step 3.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	3.14	4.42	6.21	7.42	7.71

Iteration 9: step 0.

# Insertion Sort

- Iteration  $i$ . Repeatedly swap element  $i$  with the one to its left if smaller.
- Property. After  $i$ th iteration,  $a[0]$  through  $a[i]$  contain first  $i+1$  elements in ascending order.

Array index	0	1	2	3	4	5	6	7	8	9
Value	0.32	0.56	1.12	1.17	2.78	3.14	4.42	6.21	7.42	7.71

Iteration 10: DONE.