Data Structure

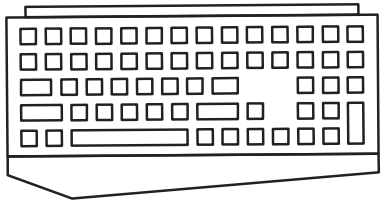
ساختمان داده

@Mscha03

Next

01

Time complexity



Next

۱. پیچیدگی زمانی کد چقدر است؟

```
1 for i in range(n):
2     print(i)
```

O(n)

Next

```python
def sum_array(arr):
    sum = 0
    for i in range(len(arr)):
        sum += arr[i]
    return sum

```

۲. پیچیدگی زمانی کد چقدر است؟

O(n)

۳. در صورتی که تعداد عناصر آرایه به دو برابر افزایش یابد چطور؟

O(n)

Next

```python
def linear_search(arr, target):
    for i in range(len(arr)):
        if arr[i] == target:
            return i
    return -1
```

۴. پیچیدگی زمانی کد چقدر است؟

O(n)

Next

```python
def factorial(n):
    return 1 if n == 0 else n * factorial(n - 1)

```

۵. پیچیدگی زمانی کد چقدر است؟ O(n)

۶. تعداد دفعات فراخوانی تابع factorial را برای یک مقدار n دلخواه مانند ۵ بررسی کنید.

۷. با تبدیل الگوریتم بازگشتی به یک نسخه غیربازگشتی، پیچیدگی زمانی آن چگونه تغییر می‌کند؟

Next

```python
def factorial(n):
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result
```

۵. پیچیدگی زمانی کد چقدر است؟ O(n)

۶. تعداد دفعات فراخوانی تابع factorial را برای یک مقدار n دلخواه مانند ۵ بررسی کنید.

۷. با تبدیل الگوریتم بازگشتی به یک نسخه غیربازگشتی، پیچیدگی زمانی آن چگونه تغییر می‌کند؟

Next

```python
for i in range(n):
    for j in range(n):
        print(i, j)
```

۸. چه تعداد عملیات چاپ انجام می‌شود؟

n^2

۹. پیچیدگی زمانی این کد چیست؟

$O(n^2)$

Next

```python
def bubble_sort(arr):
    for i in range(len(arr)):
        for j in range(0, len(arr) - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

```

تمرین ۱۰: پیچیدگی زمانی این الگوریتم را محاسبه کنید.

$O(n^2)$

تمرین ۱۱: چگونه این الگوریتم در بدترین حالت، بهترین حالت و حالت متوسط عمل می‌کند؟

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

Next

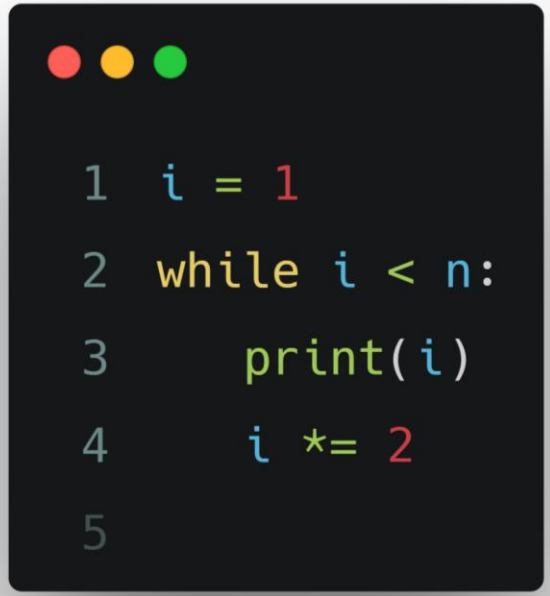
```python
def sum_matrix(matrix):
    total_sum = 0
    for i in range(len(matrix)):
        for j in range(len(matrix[i])):
            total_sum += matrix[i][j]
    return total_sum
```

۱۲. پیچیدگی زمانی این الگوریتم را تحلیل کنید.

$O(n^2)$

Next

۱۳. پیچیدگی زمانی این کد چیست؟



```
1  i = 1
2  while i < n:
3      print(i)
4      i *= 2
5
```

O (log n)

Next

```python
def binary_search(arr, target):
    low = 0
    high = len(arr) - 1
    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return -1

```

۱۴.پیچیدگی زمانی این کد چیست؟

O (log n)

۱۵. تفاوت پیچیدگی زمانی جستجوی خطی و جستجوی دودویی چقدر است؟

۱۶. اگر آرایه مرتب نباشد، آیا جستجوی دودویی همچنان کار می‌کند؟ چرا؟

Next

```python
def fib(n):
    if n <= 1:
        return n
    else:
        return fib(n-1) + fib(n-2)
```

۱۷. این کد چه کاری انجام می‌دهد؟

۱۸. پیچیدگی زمانی آن چقدر است؟

$O(2^n)$

Next

```python
def merge(arr1, arr2):
    result = []
    i, j = 0, 0
    while i < len(arr1) and j < len(arr2):
        if arr1[i] < arr2[j]:
            result.append(arr1[i])
            i += 1
        else:
            result.append(arr2[j])
            j += 1
    result.extend(arr1[i:])
    result.extend(arr2[j:])
    return result

```

۱۹. این کد چه کاری انجام می‌دهد؟

۲۰. پیچیدگی زمانی آن چقدر است؟

$O(n^2)$

Next

```python
def two_for(n):
    for i in range(n):
        print(i)
    for j in range(n):
        print(j)

```

۲۱. این کد چه کاری انجام می‌دهد؟

۲۲. پیچیدگی زمانی آن چقدر است؟

O(n)

Next

```python
for i in range(n):
    j = i * i
    while j > 0:
        j //= 4

```

۲۳. پیچیدگی زمانی آن چقدر است؟

O(n log n)

Next

```python
for i in range(1, n):
    for j in range(1, n * n, 2):
        i *= 3
```

$O(n^3)$

۲۴. پیچیدگی زمانی این کد چقدر است؟

Next

```python
def example(n):
    print(n)
    if n <= 1:
        return
    for i in range(1 , n*n):
        print("i=" , i)
    example(n-2)

```

۲۴. پیچیدگی زمانی آن چقدر است؟

$O(n^3)$

Next

بزرگ‌ترین زیربازه ۱

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک دنباله شامل n عدد صحیح مثل $a_1, a_2, \ldots, a_n$ داریم.

می‌خواهیم زیربازه‌ای از آن را انتخاب کنیم که بیشترین مجموع را دارد. به عبارت دیگر می‌خواهیم دو عدد صحیح مثل l و r که $1 \leq l \leq r \leq n$ را چنان انتخاب کنیم که مقدار $a_l + a_{l+1} + \cdots + a_r$ بیشینه باشد.

از شما می‌خواهیم برنامه‌ای بنویسید که این مقدار بیشینه را چاپ کند.

ورودی

در سطر اول ورودی عدد صحیح و مثبت n آمده است.

$$1 \leq n \leq 100$$

در سطر دوم n عدد صحیح $a_1, a_2, \ldots, a_n$ که با یک فاصله از هم جدا شده‌اند آمده است.

$$-10^9 \leq a_i \leq 10^9$$

خروجی

در تنها سطر خروجی، یک عدد صحیح که نشان‌دهنده پاسخ مسئله است را چاپ کنید.

مثال

ورودی نمونه ۱

```
6
-7 3 -1 2 -4 3
```

خروجی نمونه ۱

```
4
```

ورودی نمونه ۲

```
3
-1 -2 -3
```

خروجی نمونه ۲

```
-1
```

Next

شمارش مثلث‌ها ۱

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

تعداد سه‌تایی های نامرتب (a, b, c) را می‌خواهیم به صورتیکه بتوان مثلثی با اضلاع a و b و c ساخت و همچنین جمع آن‌ها برابر n باشد.

توجه کنید که در یک سه‌تایی نامرتب، ترتیب اعداد مهم نیست و به طور مثال سه‌تایی $(2, 3, 2)$ با سه‌تایی $(2, 2, 3)$ یکسان می‌باشد و باید یک‌بار شمرده شود.

ورودی

در تنها سطر ورودی عدد طبیعی n آمده است

$$3 \leq n \leq 100$$

خروجی

در تنها سطر خروجی باید جواب مسأله را خروجی دهید.

مثال

ورودی نمونه ۱

```
5
```

خروجی نمونه ۱

```
1
```

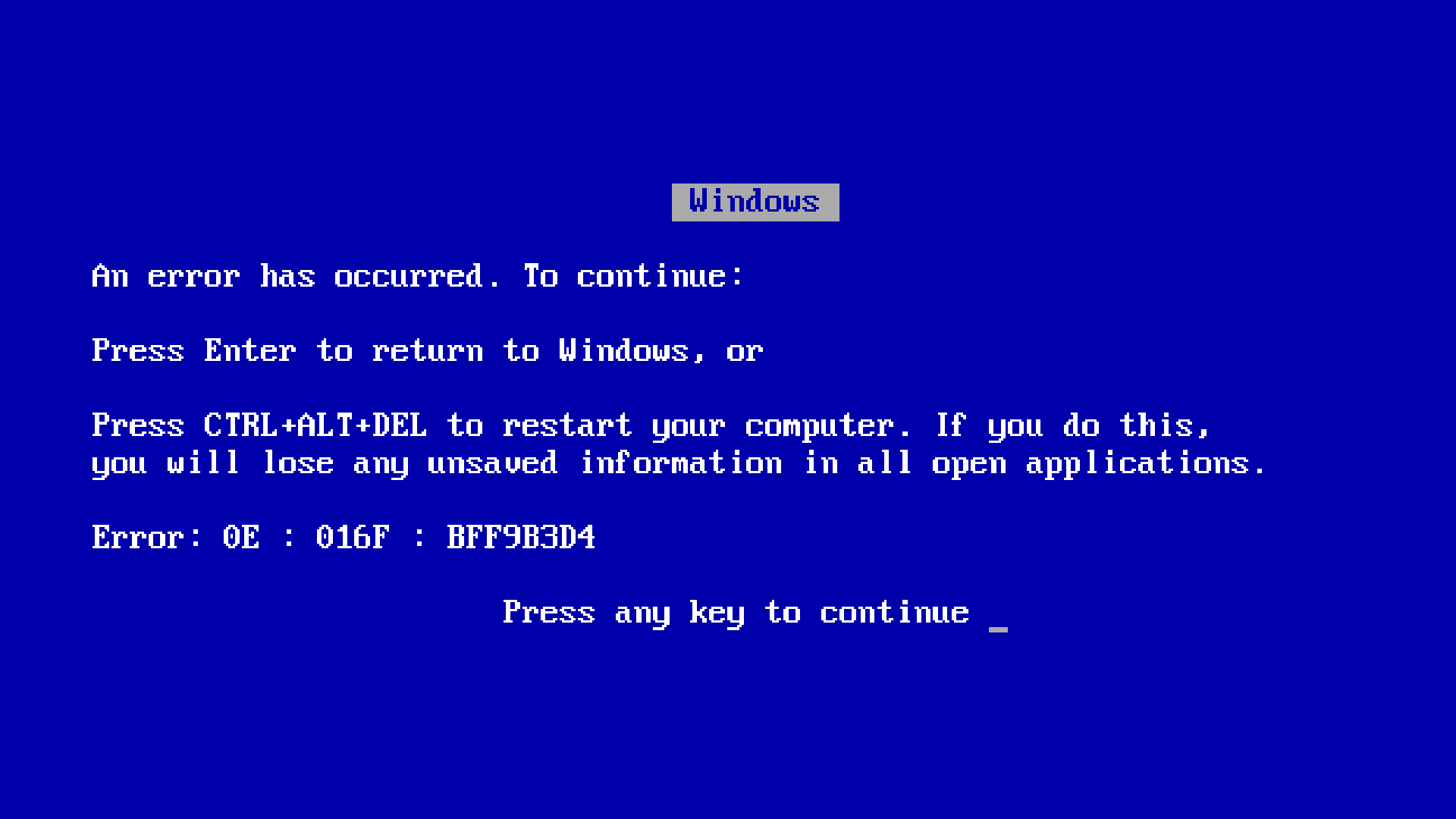
ورودی نمونه ۲

```
12
```

خروجی نمونه ۲

```
3
```

Next


Windows
An error has occurred. To continue:
Press Enter to return to Windows, or
Press CTRL+ALT+DEL to restart your computer. If you do this,
you will lose any unsaved information in all open applications.
Error: 0E : 016F : BFF9B3D4
Press any key to continue _