

Python Базовий

Робота з рекурсією

Python Базовий

Introduction



Вікторія Бойчук

Python Developer, тренер CBS

 [Вікторія Бойчук](#)



Python Базовий

Тема уроку

Робота з рекурсією

Python Базовий

План уроку

1. Що таке рекурсія у програмуванні
2. Прості варіанти застосування рекурсії
3. Бінарне дерево
4. Вирішення задач

Python Базовий

Що таке рекурсія

Рекурсія - коли функція викликає саму себе (проста рекурсія) або коли вона викликає саму себе через іншу функцію (непряма рекурсія).

Два найважливіші моменти в рекурсії: зміна стану та умова виходу.

Якщо першої умови не дотримано, код втрачає логіку, якщо друга умова не дотримана, то буде переповнення стека та помилка.

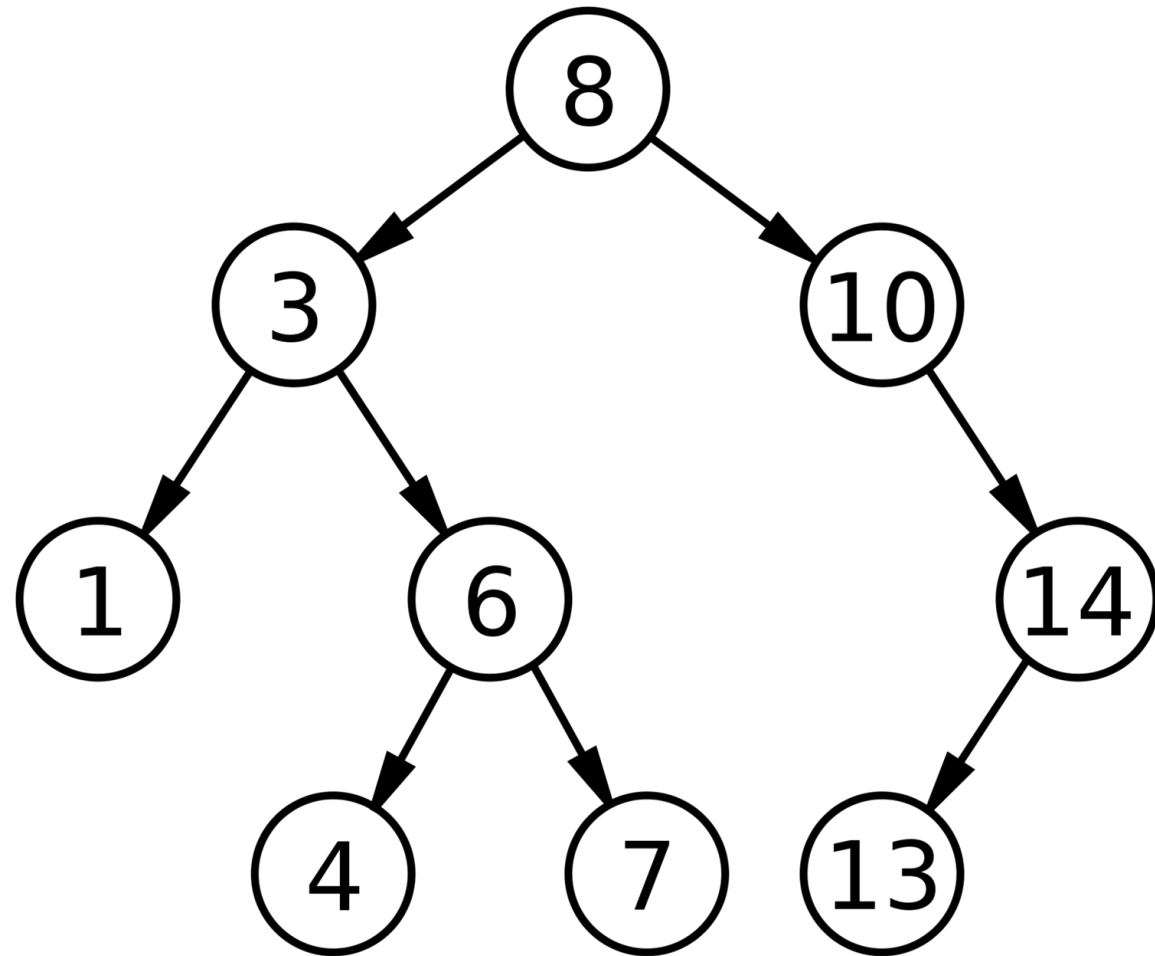
```
1      a = [1, 2, 3, 4, 5]
2
3
4      def iterate(array):
5          for elem in array:
6              print(elem)
7
8
9      def recur_iter(array, index=0):
10         if index < len(array):
11             print(array[index])
12             index += 1
13             recur_iter(array, index)
14
15
16     iterate(a)
17     recur_iter(a)
18
```

Python Базовий

Де потрібна рекурсія

Структура даних дерево - гарний приклад того, навіщо потрібна рекурсія.

Бінарне дерево — це ієрархічна структура даних, в якій кожен вузол має значення (воно ж є в даному випадку і ключем) та посилання на лівого та правого нащадка. Вузол, що знаходиться на самому верхньому рівні (який не є чиїмось нащадком) називається коренем. Вузли, які мають нащадків (обидва нащадка яких рівні NULL) називаються листям.

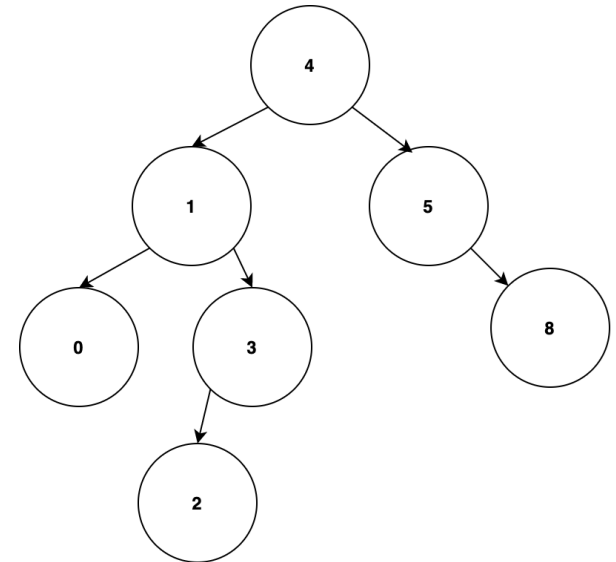


Python Базовий

Бінарне дерево на Python

```
1 class Node:
2     def __init__(self, data):
3         self.left = None
4         self.right = None
5         self.data = data
6
7     def insert(self, data):
8         if self.data:
9             if data < self.data:
10                 if self.left is None:
11                     self.left = Node(data)
12                 else:
13                     self.left.insert(data)
14             elif data > self.data:
15                 if self.right is None:
16                     self.right = Node(data)
17                 else:
18                     self.right.insert(data)
19             else:
20                 self.data = data
21
22     def print_tree(self):
23         if self.left:
24             self.left.print_tree()
25         print(self.data)
26         if self.right:
27             self.right.print_tree()
28
29
```

```
30 array = [4, 5, 1, 3, 2, 8, 0]
31 root = None
32 for i in range(len(array)):
33     if i == 0:
34         root = Node(array[i])
35     else:
36         root.insert(array[i])
37 root.print_tree()
38
```



Python Базовий

Задачі

1. Реалізувати функцію пошуку значення у бінарному дереві. Якщо значення знайдено, то функція поверне True і попередньо виведе в консоль, що значення знайдено, інакше - False і виведе в консоль, що такого значення в дереві немає.
2. Дано натуральне число N, вивести усі натуральні числа від 1 до N.
3. Дано натуральне число N. Виведіть слово YES, якщо число N є точним ступенем двійки, або слово NO у протилежному випадку.
4. Дано натуральне число N. Обчисліть суму його цифр.

Python Базовий

Відповіді

1

```
7 def search(self, data):
8     if data == self.data:
9         print("{} in tree".format(data))
10        return True
11    if data < self.data:
12        if self.left is None:
13            print("{} not in tree".format(data))
14            return False
15        return self.left.search(data)
16    if self.right is None:
17        print("{} not in tree".format(data))
18        return False
19    return self.right.search(data)
20
```

3

```
10 def check_pow_2(n):
11     if n == 1:
12         print("YES")
13     else:
14         if n % 2 == 0:
15             check_pow_2(n // 2)
16         else:
17             print("NO")
18
19 check_pow_2(128)
21
```

2

```
1 def values(n, current=1):
2     if current <= n:
3         print(current)
4         values(current=current + 1, n=n)
5
6 values(n=10)
8
```

4

```
23 def sum_val(num, res=0):
24     if not num:
25         return res
26     res += num % 10
27     num //= 10
28     return sum_val(num, res)
29
30
31 s = sum_val(125)
32 print(s)
33
```

Інформаційний відеосервіс для розробників програмного забезпечення



Перевірка знань

TestProvider.com



Перевірте як Ви засвоїли даний матеріал на [TestProvider.com](https://testprovider.com)

TestProvider – це online сервіс перевірки знань з інформаційних технологій. За його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT спеціаліста.

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.

Python Базовий

Дякую за увагу! До нових зустрічей!



Вікторія Бойчук
Python Developer, тренер CBS