



ИТ-КЛАСТЕР
СИБИРИ



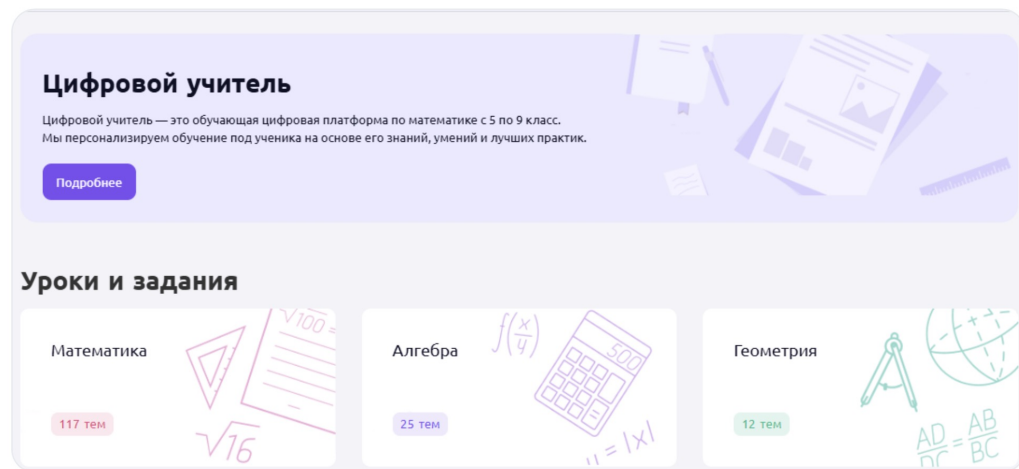
ПРЕДУНИВЕРСАРИЙ

Искусственный интеллект в контексте преподавания информатики в школе: ВЫЗОВЫ и риски

15-16 мая 2025 года, г. Омск

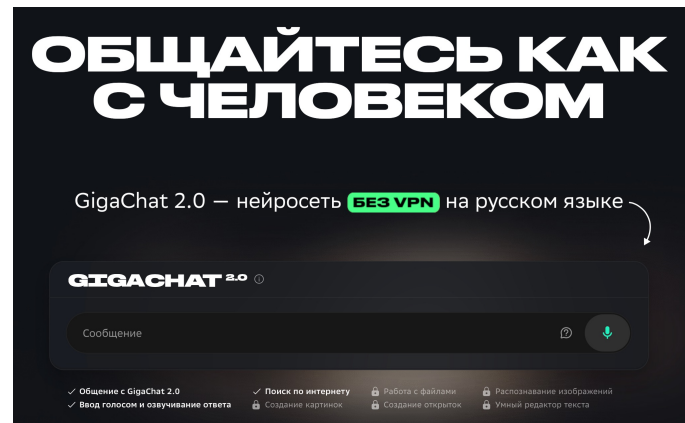
ИИ в контексте преподавании информатики в школе: вызовы и риски

Адаптация образовательного контента к уровню знаний и скорости обучения каждого учащегося при выполнении задания за счет алгоритмов, которые могут анализировать ответы, определять пробелы в знаниях и на основании этого предлагать индивидуальные учебные материалы
– это ближайшие перспективы развития ИИ для преподавания.



ИИ в контексте преподавания информатики в школе: вызовы и риски

Преподаватель информатики должен не запрещать, а стимулировать обращение учащихся к платформам ИИ в целях расширения и углубления образовательных возможностей в разных предметных областях.



Попробуйте YandexGPT 5

Попробовать YandexGPT 5 Pro в режиме чата можно в Алисе. Чтобы общаться с моделью без ограничений на число запросов, нужна опция «Алиса Про».

Открыть чат с Алисой

Бизнес-заказчикам YandexGPT 5 Pro доступна на платформе Yandex Cloud AI Studio. Там есть всё, что требуется для интеграции — API, AI Playground, инструменты для создания AI-ассистентов и другие возможности.

Использовать YandexGPT 5

Нейросеть Chat GPT

Мощный чат-бот на русском языке, разработанный OpenAI, ChatGPT отвечает на ваши вопро

⊕ Новый чат

История чатов сохраняется и доступна только для авторизованных пользователей!

Дан список чисел. Определите, есть ли такие числа в массиве, выведите их. Если ничего не выводите. Гарантируется.

Примеры

входные данные

1 2 3 -2 -4

выходные данные

ИИ в контексте преподавания информатики в школе: вызовы и риски

```
1 def find(A):
2     oppositeind = []
3     for i in range(len(A)):
4         opposite = -A[i]
5         if opposite in A[i+1:]:
6             oppositeind.append(i)
7             oppositeind.append(A.index(opposite, i + 1))
8             break
9     if oppositeind:
10        oppositeind.sort()
11        return oppositeind
12    else:
13        return None
14
15 a = input().split()
16 n = [int(num) for num in a]
17 result = find(n)
18 if result:
19     print(*result)
```

```
1 A = list(map(int, input().split()))
2 for i in range(len(A) - 1):
3     for j in range(i + 1, len(A)):
4         if A[i] + A[j] == 0:
5             print(i, j)
6             break
7
```

Однако важно понимать, что применение ИИ может потенциально создавать и угрозы

Задача №111155. Противоположные элементы в массиве

Дан список чисел. Определите, есть ли в нем два противоположных(то есть дающих в сумме 0) числа. Если такие числа есть в массиве, выведите их индексы в порядке возрастания. Если таких чисел в массиве нет, ничего не выводите. Гарантируется, что таких пар не больше одной.

Примеры

входные данные

1 2 3 -2 -4

выходные данные

1 3

```
1 A = list(map(int, input().split()))
2 i = 0
3 m = True
4 while i <= len(A) - 2 and m:
5     n = A[i]
6     for j in range(i + 1, len(A)):
7         if n + A[j] == 0:
8             print(i, j)
9             m = False
10            break
11    i += 1
```

Д. П. Кириенко. Программирование на языке Python (школа 179 г. Москвы)

[В начало](#) / [Курсы](#) / [Авторские курсы](#) / [Программирование на python](#)

Задача №3771. Родословная: подсчет уровней

 **Видеолекция по теме "Работа с файлами"**

 **Теоретический материал**

 **Задачи**

В генеалогическом древе у каждого человека, кроме родоначальника, есть ровно один родитель.

Каждому элементу дерева сопоставляется целое неотрицательное число, называемое высотой. У родоначальника высота равна 0, у любого другого элемента высота на 1 больше, чем у его родителя.

Вам дано генеалогическое древо, определите высоту всех его элементов.

Входные данные

Программа получает на вход число элементов в генеалогическом древе N . Далее следует $N - 1$ строка, задающие родителя для каждого элемента древа, кроме родоначальника. Каждая строка имеет вид **имя_потомка имя_родителя**.

Выходные данные

Программа должна вывести список всех элементов древа в лексикографическом порядке. После вывода имени каждого элемента необходимо вывести его высоту.

Примечание

Эта задача имеет решение сложности $O(n)$, но вам достаточно написать решение сложности $O(n^2)$ (не считая сложности обращения к элементам словаря).

Пример ниже соответствует приведенному древу рода Романовых.

Примеры

входные данные
9 Alexei Peter_I Anna Peter_I Elizabeth Peter_I Peter_II Alexei Peter_III Anna Paul_I Peter_III Alexander_I Paul_I Nicholaus_I Paul_I
выходные данные
Alexander_I 4 Alexei 1 Anna 1 Elizabeth 1 Nicholaus_I 4 Paul_I 3 Peter_I 0 Peter_II 2 Peter_III 2

Задача W

Тема 16

Словари и множества.

 **Видеолекция по теме "Словари и множества"**

 **Теоретический материал: множества**

 **Теоретический материал: словари**

 **Условия задач**

ИИ в контексте преподавания информатики в школе: вызовы и риски

Sum	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
25	+	+	+1	+1	+1	+	+	+3	+3	+	+6	+	+1	+7	+	+2	+1	+2	+5	+2	+1	+	+	+5	+6		
25	+	+2	+	+	+1	+3	+11	+9	+3	+1	+7	+	+3	+4	+	+1	+	+1	+3	+1	+3	+2	+1	+3	+2		
25	+	+	+9	+	+	+4	+7	+1	+	+	+	+1	+5	+2	+	+1	+	+	+2	+	+1	+2	+	+2	+2	-2	
23	+	+	+	+	+	+	+3	+1	+	+	+	+	+1	+4	+	+	+	+	+1	+4	+	-2	+	+1			
23	+	+	+3	+	+	+2	+4	+	+1	+1	+	+	+2	+1	+	+	+	+	+1	+	+	+	+1				
22	+	+	+1	+1	+	+1	+1	+1	+	+	+	+	+2	+8	+	+1	+1	+2	+	+	+	-1	+	-2	-2		
21	+1	+	+1	+3	+	+	+4	-3	+1	+	+	+	+2	+4	+	+	+2	+3	+	+1	+2	-1	+				
20	+	+	+1	+	+	+	+	+1	+	+	+	+	+	+1	+2	+1	+	+	+2	+	+						

2019-20

Sum	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
27	+	+	+	+	+	+	+2	+	+	+2	+	+	+	+3	+	+1	+1	+1	+	+	+	+3	+	+	+	+75	
26	+	+1	+	+	+	+	+4	+3	+1	+2	+	+	+	+8	+	+	+	+	+2	+	+	+	+	+	+	+36	-6
26	+	+	+1	+	+1	+	+	+	+	+	+	+	+	+	+1	+	+1	+	+2	+	+1	+	+1	+	+	+	-4
26	+	+	+1	+1	+2	+	+2	+	+1	+1	+	+	+	+	+	+	+11	+2	+	+	+	+1	+	+2	+4		
25	+	+	+1	+	+	+	+1	+1	+	+	+	+	+2	+2	+1	+	+	+	+	+	+	+1	+1	+1	+	-8	-3
25	+	+	+1	+	+	+	+1	+1	+1	+	+	+	+	+5	+	+	+3	+2	+	+6	+	+3	+	+3	+1	+	
25	+	+1	+2	+	+1	+	+6	+1	+	+3	+	+	+1	+	+1	+1	+4	+	+1	+10	+4	+6	+	+1	+		
22	+	+	+1	+	+	+3	+2	+	+	+2	+	+	+	+5	+	+	+	+2	+	+5	+	+3	+				
22	+	+5	+2	+	+	+5	+4	+1	+2	+	+4	+	+	+	+3	+	+3	+2	+9	+7	+7	+	+	+1			
22	+1	+	+	+1	+	+	+	+2	+	+2	+	+	+	+	+1	+	+1	+	+	+1	+3	+	+	+1			
22	+1	+	+1	+	+1	+	+	+1	+5	+	+1	+2	+3	+	+	+	+2	+2	+2	+	+	+	+	+3			
20	+2	+	+	+	+2	+2	+2	+	+	+1	+	+	+	+	-3	+	+	+	+1	+2	+	+	+				

2024-25

Sum	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
25	+	+	+1	+	+1	+1	+	+1	+2	+	+4	+	+1	+1	+	+	+	+	+7	+5	+1	+2	+	+	+		
25	+2	+2	+5	+	+1	+7	+	+5	+	+	+	+	+	+2	+13	+3	+	+1	+3	+13	+1	+	+12	+	+1		-25
25	+	+4	+6	+	+5	+1	+10	+11	+1	+3	+	+	+	+	+	+1	+4	+	+	+15	+1	+1	+	+1	+11		-4
24	+1	+	+	+	+	+	+	+1	+	+3	+1	+	+	+	+2	+1	+3	+1	+1	+	+	+	+	+	+		-1
24	+1	+	+	+	+	+	+	+2	+	+	+	+	+	+	+3	+	+	+	+	+1	+2	+1	+3	+1	+1		-1
24	+	+	+	+	+	+1	+2	+1	+	+	+1	+1	+	+1	+2	+	+	+	+	+4	+	+	+	+3	+1	-2	
23	+	+	+	+	+	+	+	+1	+	+	+	+3	+	+2	+2	+	+	+	+	+	+1	+	+	+	+1		-2
23	+	+	+	+	+	+	+	+2	+5	+2	+1	+	+	+2	+8	+	+6	+	+	+3	+2	+2	+	+1	+		-6
23	+1	+	+3	+1	+	+	+	+	+	+	+	+1	+	+	+	+	+1	+	+1	+2	+8	+	+	+	+	-4	
23	+	+	+1	+	+	+	+	+	+	+	+2	+	+	+1	+1	+	+	+	+	+3	+5	+3	+	+	+3		
22	+	+	+3	+1	+	+	+	+2	+2	+3	+2	+	+	+1	+1	+	+1	+2	+	+	+2	+6	+	-6	+		
22	+	+	+1	+	+	+	+	+	+1	+	+	+	+	+	+7	+	+1	+1	+1	+	+5	+1	+	+	+3	-2	
21	+	+	+3	+	+1	+4	+2	+	+	+	+	+	+	+	+2	+5	+	+	+	+	+	+	+1	+	+	-1	
21	+	+1	+	+	+1	+1	+2	+	+2	+1	+1	+	+	+1	+7	+5	+3	+1	+	+2	+2	+4	+	+	+		
20	+1	+	+2	+	+	+	+	+	+	+2	+2	+	+4	+	+	+1	+2	+	+	+14	+2	+2	-1	+	+		
20	+	+	+2	+	+1	+1	+	+	+	+	+	+	+	+	+	+3	+10	+1	+	+	-2	+	+	-5			
20	+	+	+1	+	+1	+1	+	+	+5	+5	+	+4	+	+	+1	+1	+	+	+	+	+	+	+	+	+		
20	+	+1	+2	+	+	+	+	+	+1	+1	+1	+1	+1	+5	+	+3	+2	+2	+	+12	+7	+	+	+	-2		
20	+	+	+1	+2	+	+	+7	+9	+1	+1	+1	+	+	+2	+7	+	+1	+1	+1	+	+7	+3	-4	+1	-3		
20	+1	+1	+1	+1	+	+	+4	+	+	+	+2	+	+	+	+	+	+	+	+	+	+1	+	+	+	+		
20	+	+	+	+3	+2	+	+5	+	+	+3	+	+2	+1	+	+	+	+4	+	+	+	+	+	+	+	+		-2

2023-24

ИИ в контексте преподавания информатики в школе: вызовы и риски

```
def calculate_height(name, parent_to_child_map, heights):  
    """Рассчёт уровня каждого члена семейного дерева: param name: Имя члена семьи: param  
    parent_to_child_map: Словарь, отображающий связь родитель->ребёнок: param heights: Словарь высот,  
    куда записывается уровень каждого члена."""  
    if name not in heights:  
        # Если родитель известен, рассчитываем уровень рекурсивно  
        if name in parent_to_child_map:  
            parent_name = parent_to_child_map[name]  
            height = calculate_height(parent_name, parent_to_child_map, heights) + 1  
        else:  
            # Корень дерева, уровень 0  
            height = 0  
        heights[name] = height  
    return heights[name]  
  
# Чтение входных данных  
N = int(input()) # Читаем количество записей  
parent_to_child_map = {} # Хранит карту родитель-потомок  
for _ in range(N - 1):
```

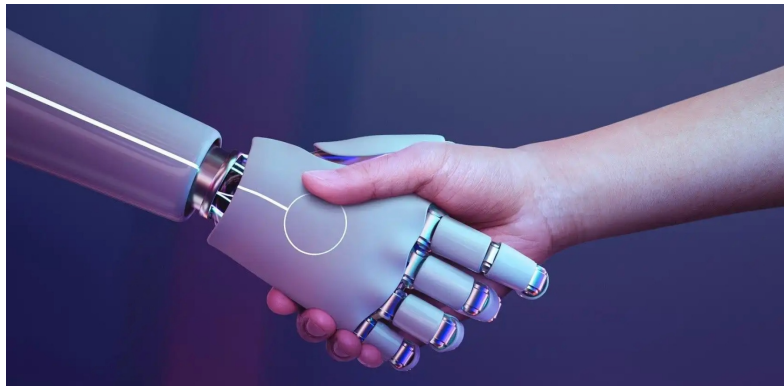
```
    child, parent = input().strip().split()  
    parent_to_child_map[child] = parent  
    heights = {} # Здесь будут храниться уровни каждого человека  
    # Вычисление уровней для всех членов семей  
    for person in parent_to_child_map.keys():  
        calculate_height(person, parent_to_child_map, heights)  
    # Добавляем уровень корня вручную  
    root_person = next(iter(set(parent_to_child_map.values()) - set(parent_to_child_map.keys())))  
    heights[root_person] = 0  
    # Выведем элементы в лексикографическом порядке  
    sorted_people = sorted(heights.keys())  
    for person in sorted_people:  
        print(f'{person} {heights[person]}')
```

ИИ в контексте преподавании информатики в школе: вызовы и риски

```
1 def main():
2     N = int(input())
3     connections = dict()
4     for _ in range(N - 1):
5         child, parent = input().strip().split()
6         connections[child] = parent
7     all_children = set(connections.keys())
8     all_parents = set(connections.values())
9     root = next(iter(all_parents - all_children))
10    levels = {root: 0}
11
12    def dfs(current_person):
13        children = [k for k, v in connections.items() if v == current_person]
14        for child in children:
15            levels[child] = levels[current_person] + 1
16            dfs(child)
17
18    dfs(root)
19
20    names = sorted(levels.keys())
21    for name in names:
22        print(f"{name} {levels[name]}")
23
24    main()
```

```
1 def par(d, name):
2     if name not in d or d[name] == None:
3         return 0
4     return 1 + par(d, d[name])
5
6 f = open('input.txt', 'r', encoding='utf-8')
7 k = int(f.readline())
8 d = {}
9 for i in f:
10    j = i.split()
11    name, parent = j[0], j[1]
12    d[name] = parent
13    if parent not in d:
14        d[parent] = None
15 d1 = sorted(d.keys())
16 for i in d1:
17    print(i, par(d, i))
```

ИИ в контексте преподавания информатики в школе: вызовы и риски



Спасибо за внимание

Александр Дятлов