Базовые структуры данных

Михаил Каменщиков

Повторение

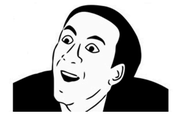
- У меня есть код, как оценить его сложность?
- Что такое О-большое?
- Я посчитал сложность алгоритма, как мне понять, какой выбрать?

План занятия

- Выясним, как данные можно хранить в памяти компьютера, рассмотрим модель массива и связного списка
- Поищем цикл в связном списке
- Узнаем, что такое абстрактный тип данных на примере очереди и стека
- Решим задачку на валидацию скобочной последовательности
- Напишем свою очередь, оценим сложность операций

Структуры данных

- Структуры данных нужны для хранения данных



- Структуры данных умеют что-то делать эффективно, а что-то не очень
- Выбор структуры зависит от требований задачи, универсальной структуры данных не существует
- Какие структуры вы чаще всего используете в работе?

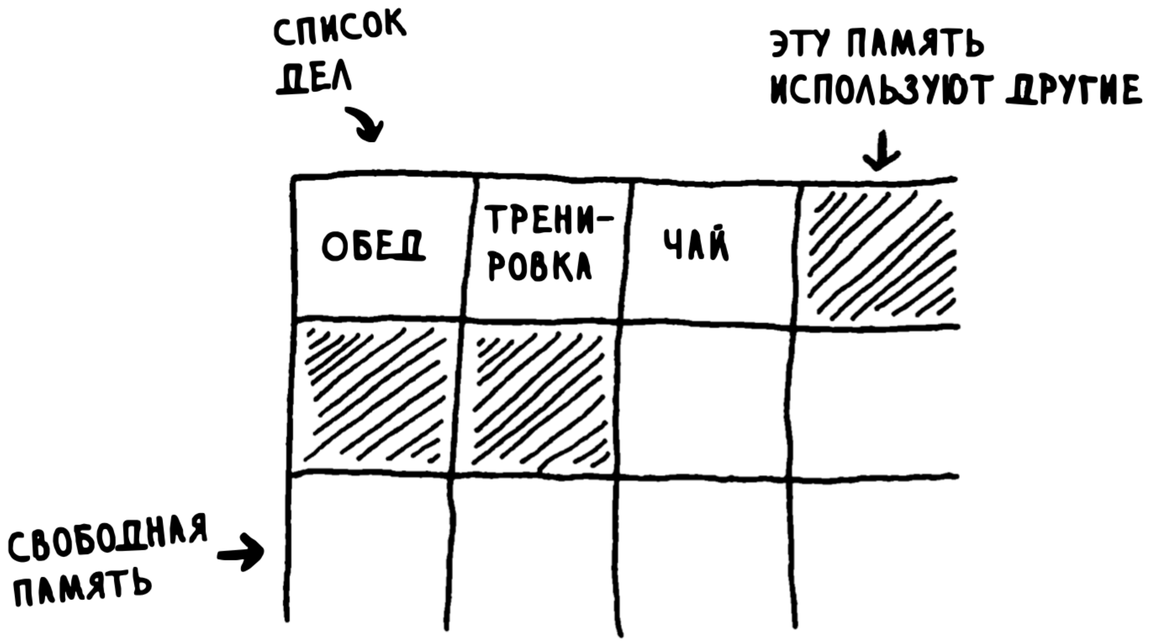
Массив

- Непрерывный участок в памяти компьютера
- Доступна арифметика адресов
- Быстрый доступ по индексу
- Медленная вставка/удаление из произвольного места
- List в Питоне - это массив!

Амортизированная стоимость

- Если у нас есть некоторая «тяжелая» операция, которая выполняется достаточно редко, и много «легких» операций
- То можно «размазать» стоимость большой операции по маленьким и получить таким образом оценку сложности
- Пример: добавление элемента в конец динамического массива

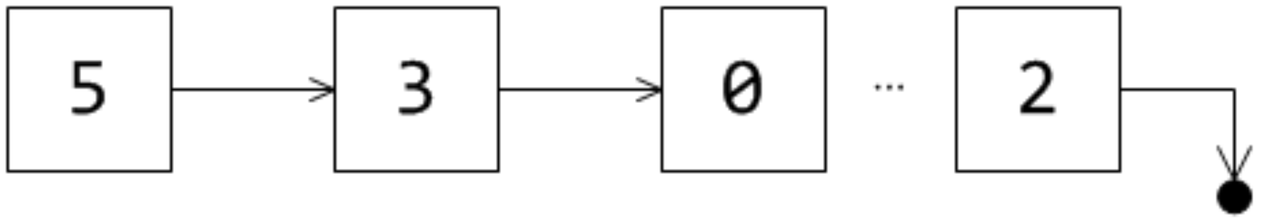
Недостаток массива



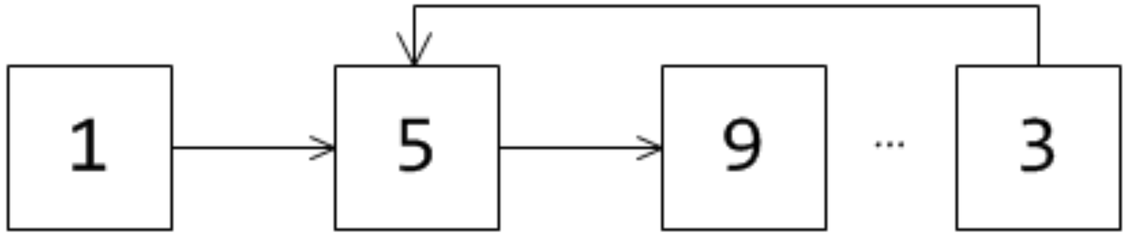
Связный список

- Коллекция элементов, где каждый элемент содержит указатель на следующий
- Эффективное добавление элементов
- Неэффективный доступ к произвольному индексу
- Требует больше памяти, чем массив, т.к. нужно хранить указатели
- https://visualgo.net/en/list

Поиск цикла в связном списке



Normal linked list



Linked list with a cycle

Перерыв

Сложность операций

	Массив	Связный список
Доступ к i-му элементу		
Удаление элемента		
Вставка в середину		
Вставка в конец		

Абстрактный тип данных

- Некоторая сущность, обладающий заданным набором свойств
- Пример из математики: целые числа, свойства - арифметические операции и операции сравнения
- Реализация скрыта от пользователя
- Как устроены целые числа в C? а в Python?

Стек

Стек

- Абстрактный тип данных с операциями $push, peek, pop$
- $push$ - добавить элемент на вершину стека
- $peek$ - получить значение элемента на вершине стека
- pop - удалить элемент с вершины стека
- Какая структура данных подходит для реализации?

Скобочная последовательность

- Представим, что нам нужно реализовать статический анализ кода в IDE
- Начнем с того, чтобы указывать на ошибку в количестве скобок
- Как решить эту задачу при помощи стека?
- А если видов скобок несколько?

Очередь

Очередь

- Абстрактный тип данных с операциями *enqueue*, *dequeue*, *peek*
- *enqueue* - добавить элемент в конец очереди
- *dequeue* - получить значение из начала очереди
- *peek* - «подсмотреть» элемент в начале очереди
- Какая структура данных подходит для реализации?

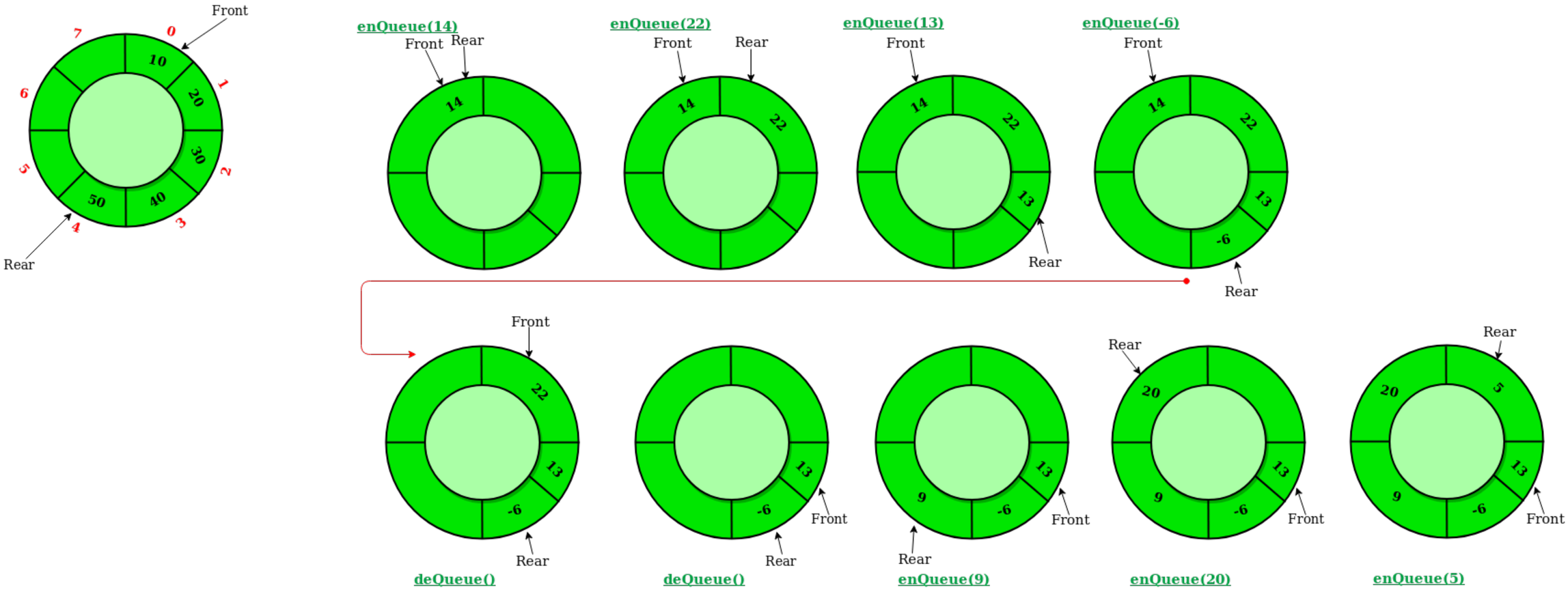
Очередь из двух стеков

- Есть 2 стека, нужно реализовать очередь с помощью них
- Как это сделать?

Очередь из двух стеков

- Добавление элемента - понятно
- Удаление элемента - ???

Очередь из массива



https://www.cs.usfca.edu/~galles/visualization/QueueArray.html

Выводы

- Чем массив лучше связного списка? Чем хуже?
- В каких задачах будет полезен стек?
- В каких задачах будет полезна очередь?