

Алгоритмы и структуры данных

Лекция 2

Массивы, списки, очереди

Кандауров Геннадий



education

Напоминание отметиться на портале

+ ОСТАВИТЬ ОТЗЫВ

vk образование

БлогиЛюдиПрограммаВакансииРасписание

Q<

VK

Техно

Открыт приём заявок!

чт, 8 сентября

Нет занятий

пт, 9 сентября

18:00 Углубленный Py... с3
Введение в Python, основные
понятия, тестирование
Г. Кандауров

сб, 10 сентября

Нет занятий

вс, 11 сентября

Нет занятий

пн, 12 сентября

Нет занятий

Углубленный Python

↓ 0 ↑

Блог для материалов по курсу "Углубленный Python"

57 читателей, 2 топика

ПодписатьсяСоздать топик

Поиск по авторам, заголовку и тексту топика...

Найти

Добро пожаловать на курс!

Углубленный Python

ИзменитьУдалить

Всем привет и добро пожаловать на курс по углубленному изучению Python!

Прямой эфир

МоиВсе

Геннадий Кандауров час назад
Углубленный Python → Добро пожаловать
на курс! 0

Екатерина Черкасова 7 дней назад
Стажировка → Приглашаем мобильных,
фронтенд- и бэкэнд-разработчиков на
Weekend Offer! 0

Дарья Вовченко 9 дней назад
Углубленный Python → Добро пожаловать
в образовательные проекты VK
Образование! 0

Дарья Вовченко 9 дней назад
Разработка веб-сервисов на

Содержание занятия

1. Массивы
2. Динамические массивы
3. Односвязные и двусвязные списки
4. Очереди и стеки
5. Двусвязная очередь
6. Подходы к решению задач

Массивы и списки



Массив

Массив – набор однотипных элементов, расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексам.

Традиционно индексирование элементов массивов начинают с 0.

Размерность массива – количество индексов, необходимое для однозначного доступа к элементу массива.

[1, 2, 9, 5, 3, 2]

0 1 2 3 4 5

Динамический массив

Динамическим называется **массив**, размер которого может изменяться во время исполнения программы.

Динамический массив содержит внутренний массив фиксированной длины для хранения элементов. Внутренний массив называется буфером. Помнит текущее количество добавленных элементов. Размер буфера имеет некоторый запас для возможности добавления новых элементов.

[1, 2, 9, 5, 3, 2, ..., ...] # 6 в буфере и 2 свободных слота
0 1 2 3 4 5 6 7

Динамический массив

Буфер может закончиться.

Если буфер закончился, то при добавлении нового элемента:

1. выделяется новый буфер, больший исходного;
2. копируется содержимое старого буфера в новый;
3. добавляется новый элемент.

[1, 2, 9, 5, 3, 2, 4, 4] # добавляем 0, но места нет

0 1 2 3 4 5 6 7

[1, 2, 9, 5, 3, 2, 4, 4, 0, ..., ..., ..., ..., ..., ..., ..., ...]

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Динамический массив

Как долго работает метод `append()` добавления элемента?

- В лучшем случае = $O(1)$
- В худшем случае = $O(n)$
- В среднем?

Имеет смысл рассматривать несколько операций добавления и оценить среднее время в контексте последовательности операций.

Подобный анализ называется **амортизационным**.

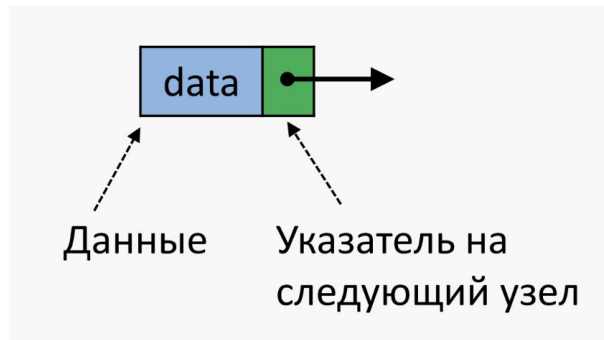
Связные списки

Связный список — динамическая структура данных, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну или две ссылки («связки») на следующий и/или предыдущий узел списка.

Преимущество перед массивом:

- Порядок элементов списка может не совпадать с порядком расположения элементов данных в памяти, а порядок обхода списка всегда явно задаётся его внутренними связями.

Односвязный список



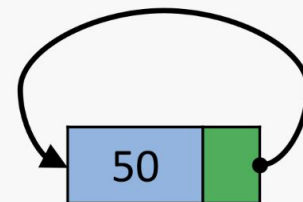
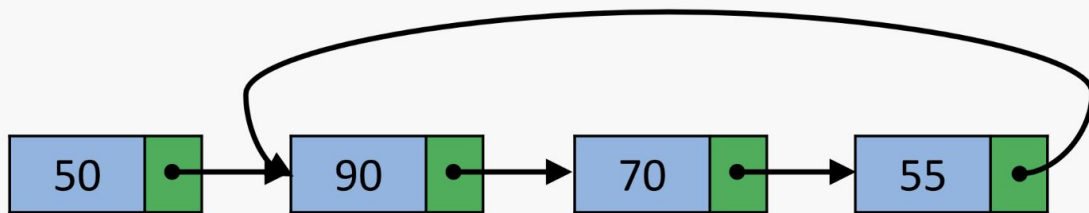
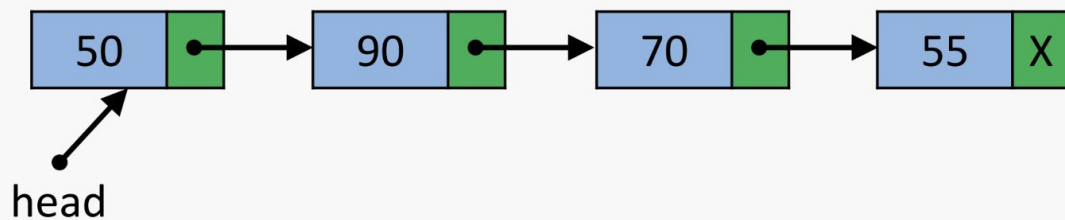
```
class Node:  
    def __init__(self, val):  
        self.val = val  
        self.next = None
```

Ссылка в каждом узле одна.

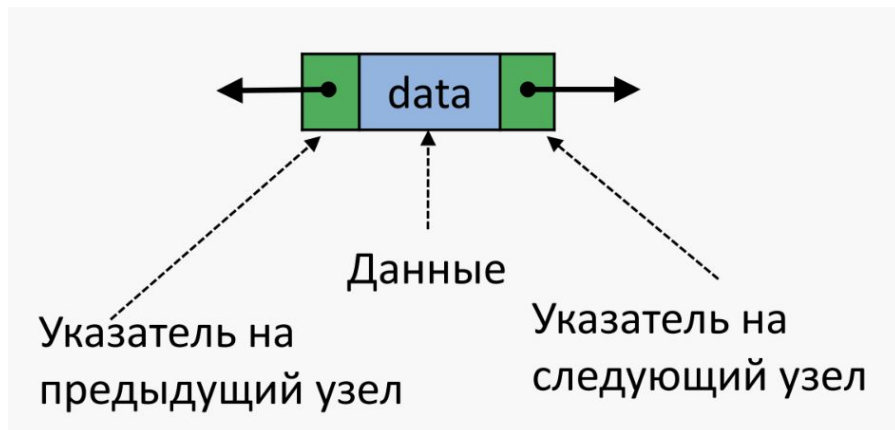
Указывает на следующий узел в списке.

Узнать адрес предыдущего элемента, опираясь на содержимое текущего узла, невозможно.

Односвязный список

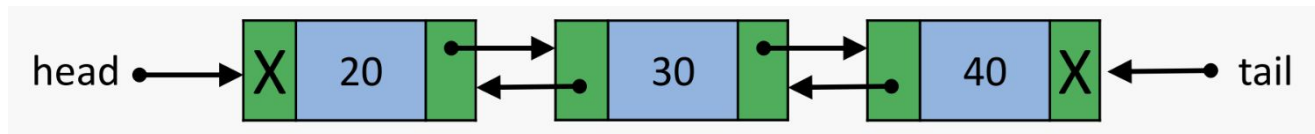


Двусвязный список



```
class Node:
    def __init__(self, val):
        self.val = val
        self.next = None
        self.prev = None
```

Указатель в каждом узле указывают на предыдущий и на последующий узел в списке.



Связные списки

Операции со списками:

- Поиск элемента
- Вставка элемента
- Удаление элемента
- Объединение списков

Сравнение списков и массивов

	Массив	Односвязный список
Вставка	$O(n)$	$O(1)$
Добавление (вставка в конец)	$O(1)$	$O(n)/O(1)^*$
Удаление	$O(n)$	$O(1)$
Доступ по индексу	$O(1)$	$O(n)$

Сравнение списков и массивов

Недостатки списков

- Нет быстрого доступа по индексу
- Расходуется память на указатели
- Узлы могут располагаться в памяти разреженно, что не позволяет использовать кэширование процессора

Преимущества списков

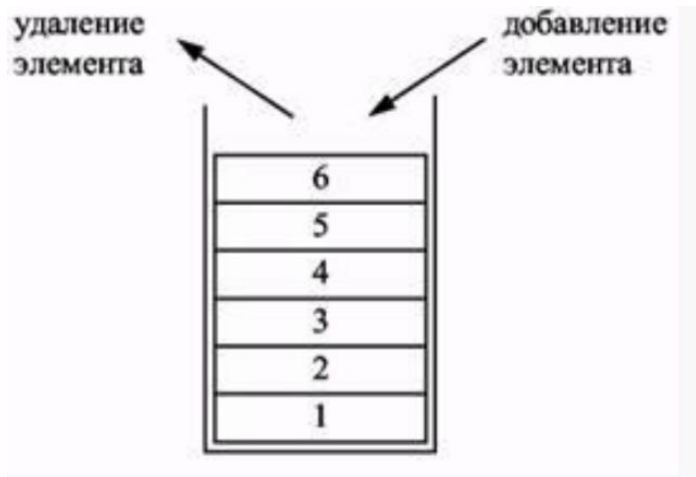
- Быстрая вставка узла
- Быстрое удаление узла
- Не тратим доп. память по сравнению с динамическим массивом

Стек

Стек – абстрактный тип данных (или структура данных), представляющий из себя список элементов, организованный по принципу LIFO = Last In First Out, «последним пришел, первым вышел».

Операции:

- Вставка (Push)
- Извлечение (Pop) – извлечение элемента, добавленного последним.

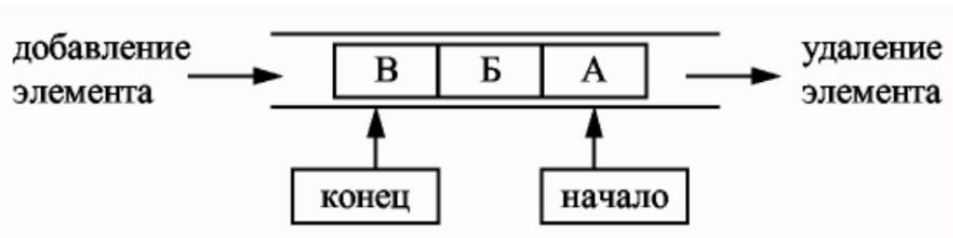


Очередь

Очередь – абстрактный тип данных (или структура данных), представляющий из себя список элементов, организованный по принципу FIFO = First In First Out, «первым пришел, первым вышел».

Операции:

- Вставка (Enqueue)
- Извлечение (Dequeue) – извлечение элемента, добавленного первым.

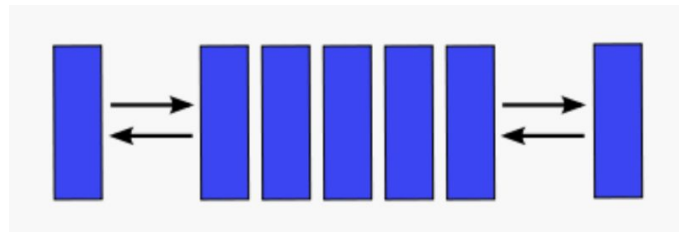


Двусвязная очередь

Двусвязная очередь (дэк, deque) – абстрактный тип данных (структура данных), в которой элементы можно добавлять и удалять как в начало, так и в конец, то есть принципами обслуживания являются одновременно FIFO и LIFO.

Операции:

- Вставка в конец (PushBack)
- Вставка в начало (PushFront)
- Извлечение из конца (PopBack)
- Извлечение из начала (PopFront)



Массивы

В Python:

- `list`
- `tuple`
- `array`
- `queue`
- `deque`

Подходы к решению задач

- Метод двух указателей
- Метод скользящего окна

Напоминание отметиться на портале Vol 2

+ ОСТАВИТЬ ОТЗЫВ ПОСЛЕ ЛЕКЦИИ

vk образование

БлогиЛюдиПрограммаВакансииРасписание

Q<

VK

Техно

Открыт приём заявок!

чт, 8 сентября

Нет занятий

пт, 9 сентября

18:00 Углубленный Py... с3
Введение в Python, основные
понятия, тестирование
Г. Кандауров

сб, 10 сентября

Нет занятий

вс, 11 сентября

Нет занятий

пн, 12 сентября

Нет занятий

Углубленный Python

↓ 0 ↑

Блог для материалов по курсу "Углубленный Python"

57 читателей, 2 топика

Подписаться

Создать топик

Поиск по авторам, заголовку и тексту топика...

Найти

Добро пожаловать на курс!

Углубленный Python

Изменить

Удалить

Всем привет и добро пожаловать на курс по углубленному изучению Python!

Прямой эфир

МоиВсе

Геннадий Кандауров час назад
Углубленный Python → Добро пожаловать
на курс! 0

Екатерина Черкасова 7 дней назад
Стажировка → Приглашаем мобильных,
фронтенд- и бэкэнд-разработчиков на
Weekend Offer! 0

Дарья Вовченко 9 дней назад
Углубленный Python → Добро пожаловать
в образовательные проекты VK
Образование! 0

Дарья Вовченко 9 дней назад
Разработка веб-сервисов на

Спасибо за
внимание

