ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Департамент программной инженерии

Алгоритмы и структуры данных

Семинар №6. 2021-2022 учебный год

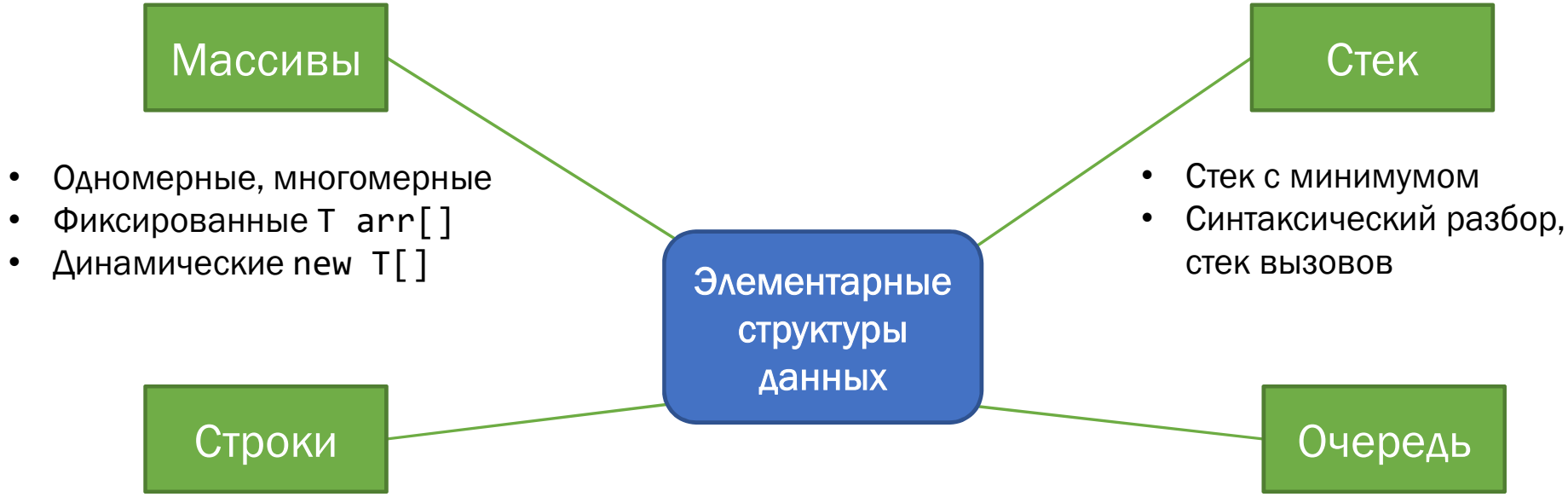
Нестеров Роман Александрович, *ДПИ ФКН и НУЛ ПОИС*

Бессмертный Александр Игоревич, *ДПИ ФКН*

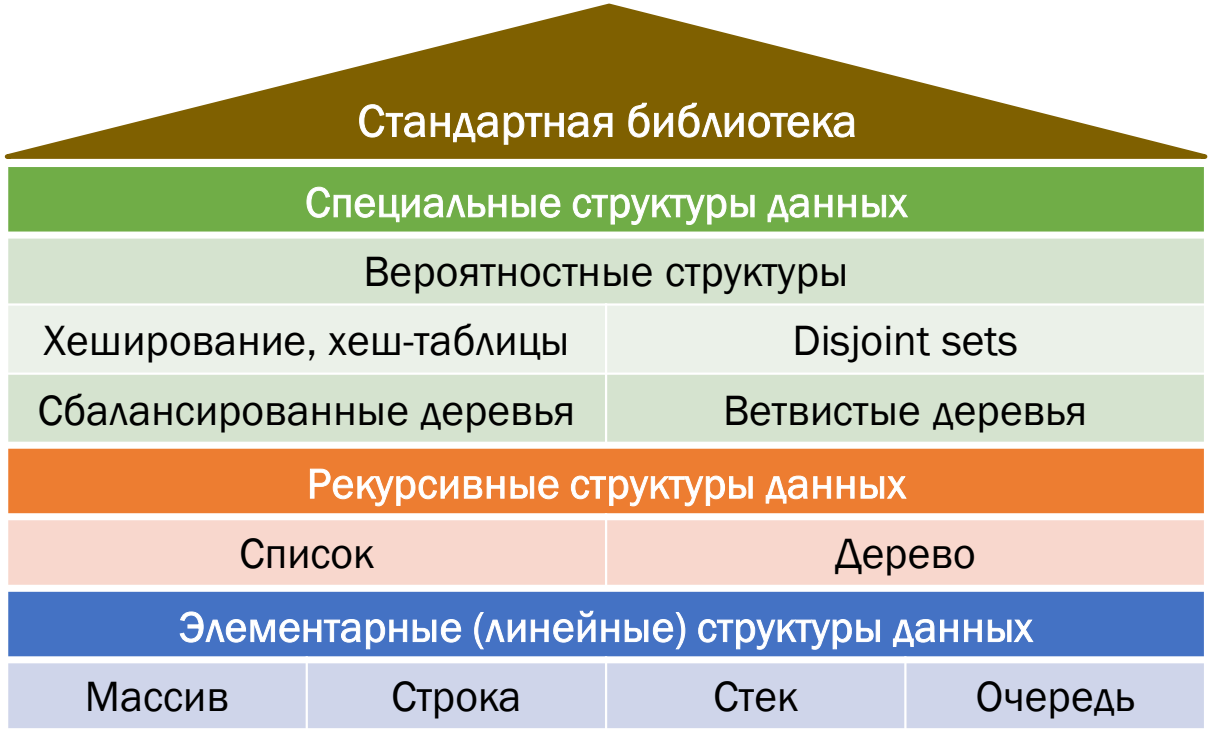
Higher School of Economics Faculty of Computer science

Das Billigste ist immer das Teuerste

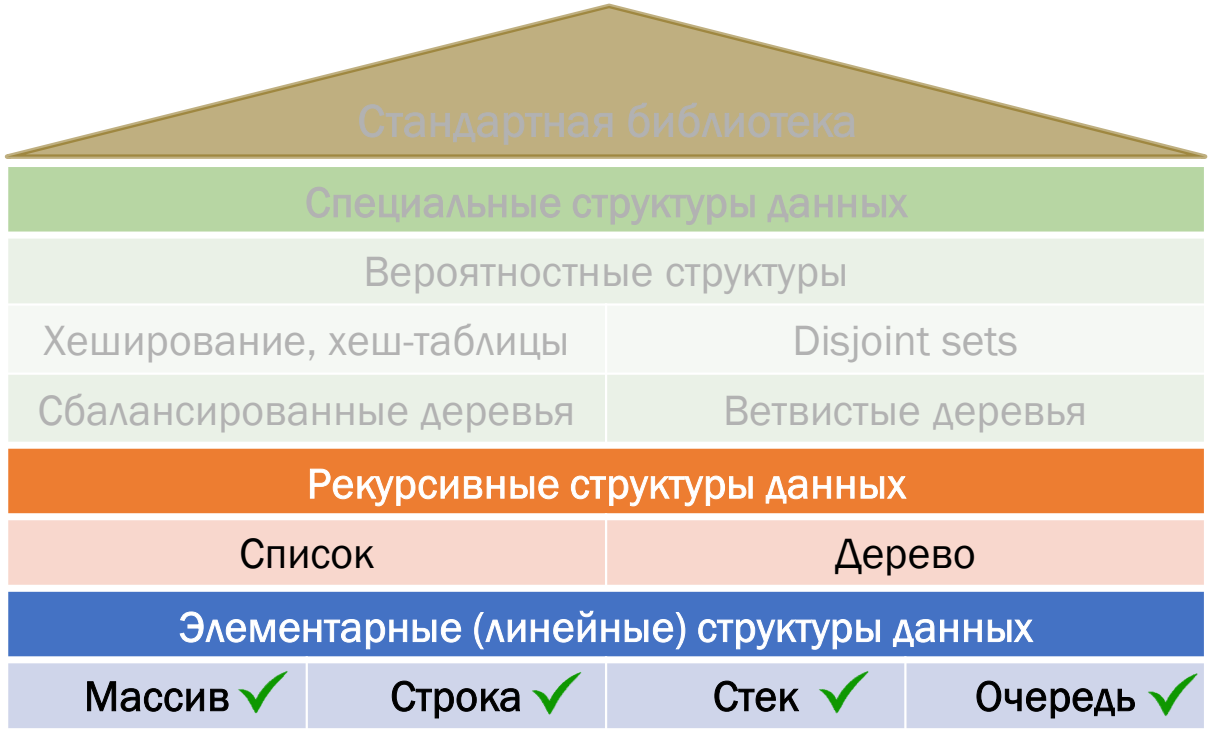
Recap


Элементарные структуры данных
Массивы
Одномерные, многомерные
Фиксированные T arr[]
Динамические new T[]
Стек
Стек с минимумом
Синтаксический разбор, стек вызовов
Строки
Массивы символов char[]
Объекты std::string
Очередь
Одно/двухсторонняя
Кольцевой массив
Очередь на двух стеках

Где мы?


Стандартная библиотека
Специальные структуры данных
Вероятностные структуры
Хеширование, хеш-таблицы
Disjoint sets
Сбалансированные деревья
Ветвистые деревья
Рекурсивные структуры данных
Список
Дерево
Элементарные (линейные) структуры данных
Массив
Строка
Стек
Очередь

Где мы?


Стандартная библиотека
Специальные структуры данных
Вероятностные структуры
Хеширование, хеш-таблицы
Disjoint sets
Сбалансированные деревья
Ветвистые деревья
Рекурсивные структуры данных
Список
Дерево
Элементарные (линейные) структуры данных
Массив
Строка
Стек
Очередь

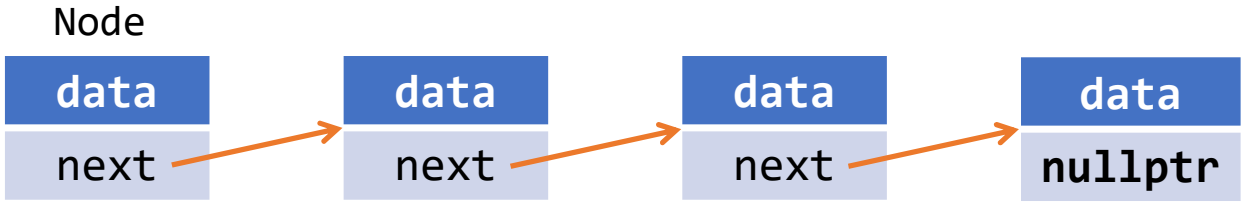
План

- Односвязный список
- Двусвязный список
- Циклический список
- Развернутый (unrolled) список
- Реализация

Односвязный список

Односвязный список

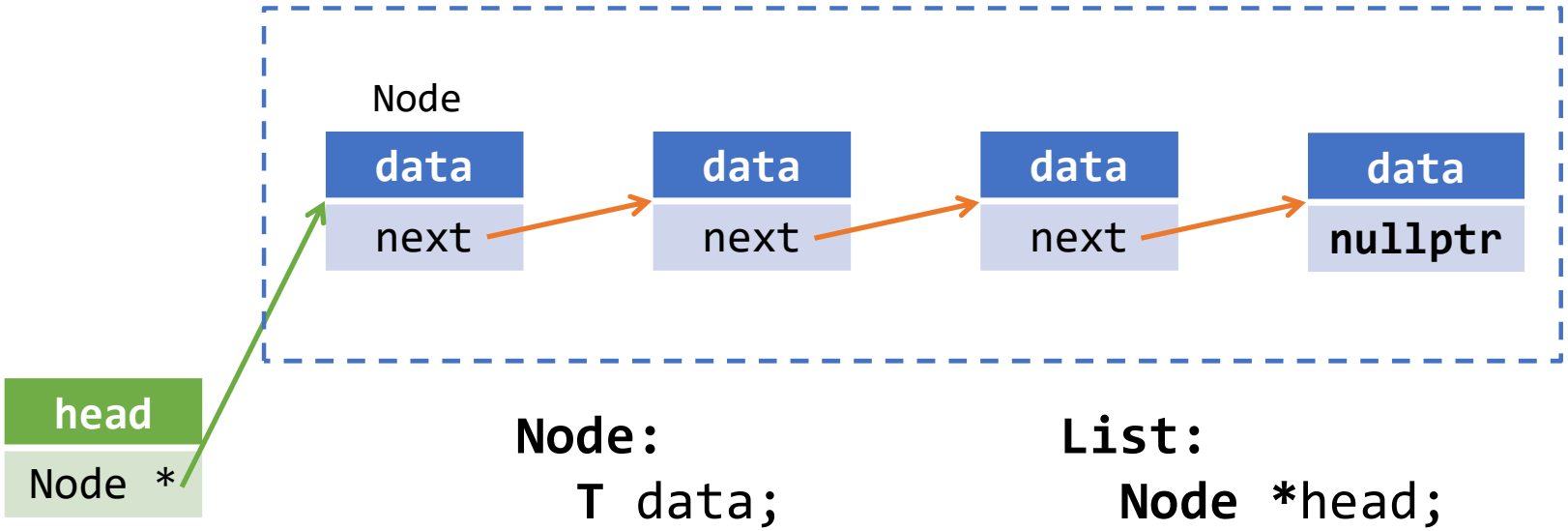
Указатель `next` указывает на следующий узел того же типа.




```
Node:
   T data;
   Node *next;
```

Односвязный список

Указатель `next` указывает на следующий узел того же типа.



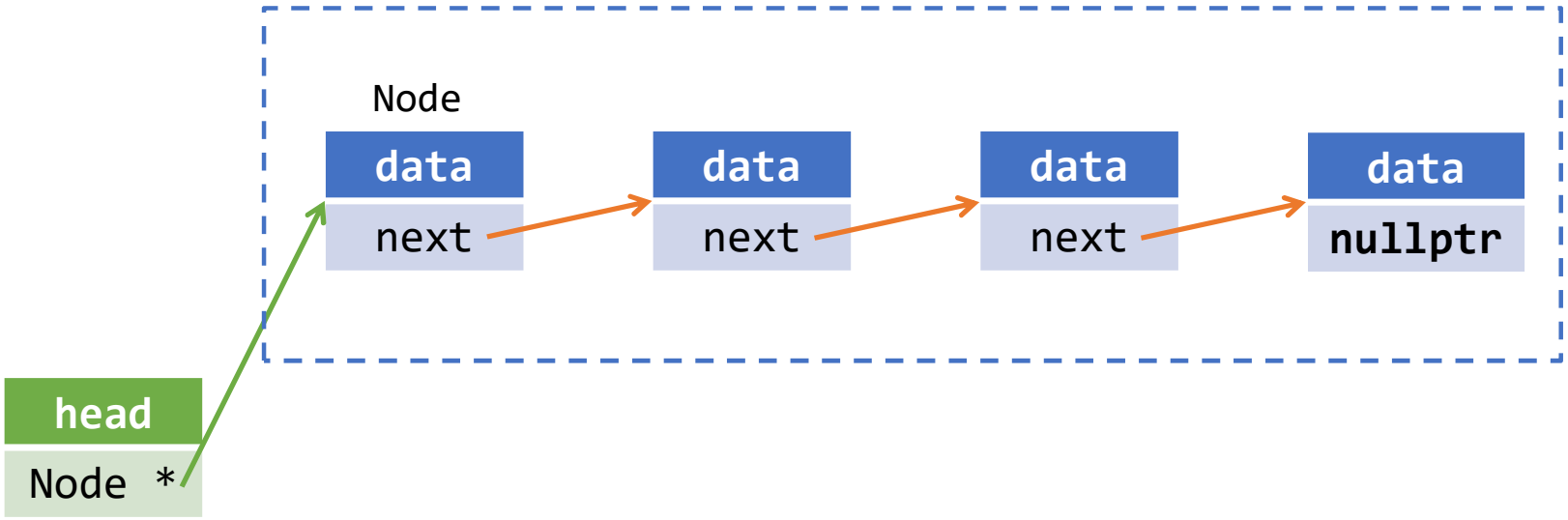

```
Node:
  T data;
  Node *next;
```

```
List:
  Node *head;
  List();
```

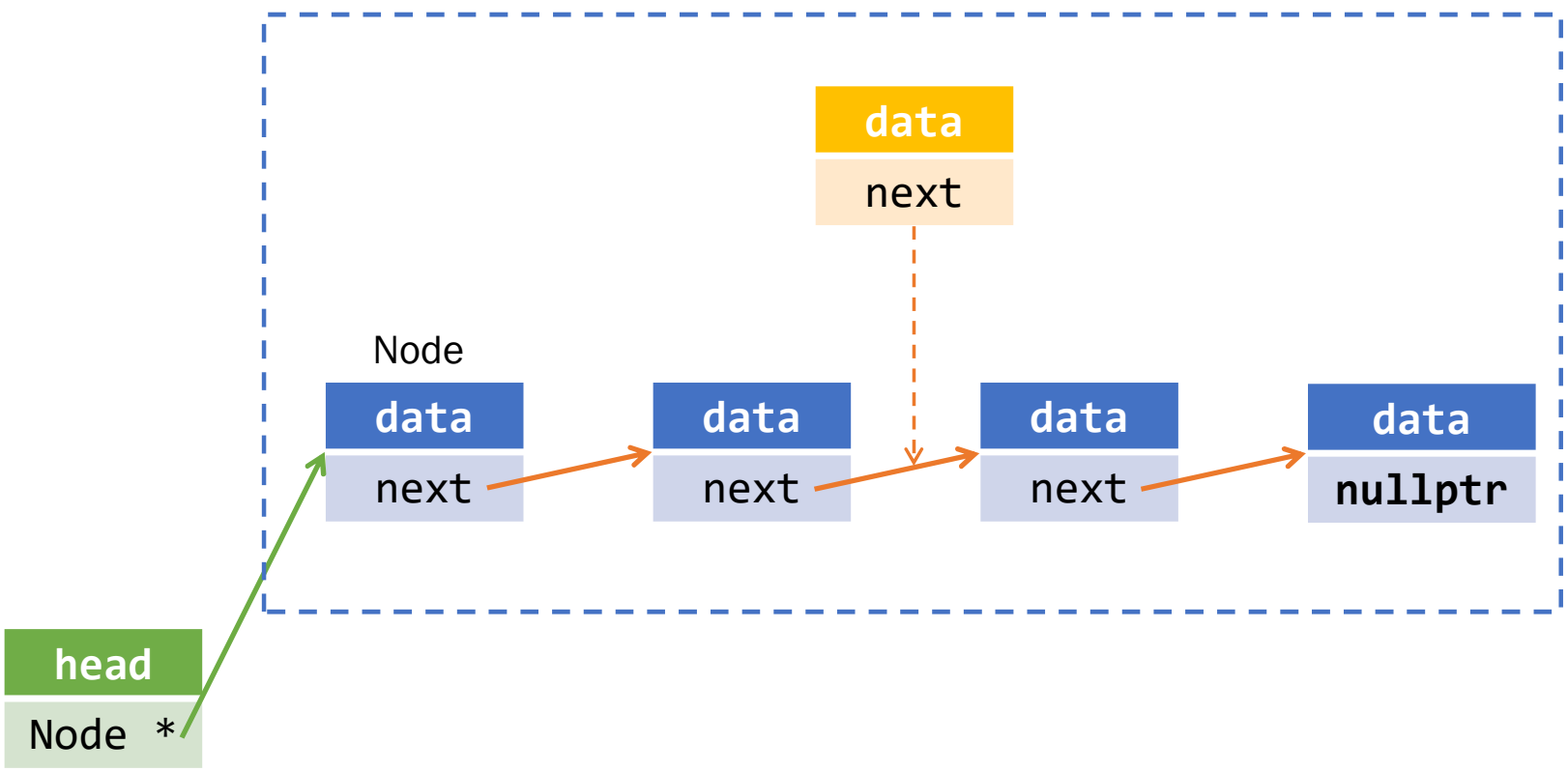
Односвязный список. Базовые операции

- Обход списка
- Поиск узла (по ключу/по индексу)
- Вставка узла
- Удаление узла

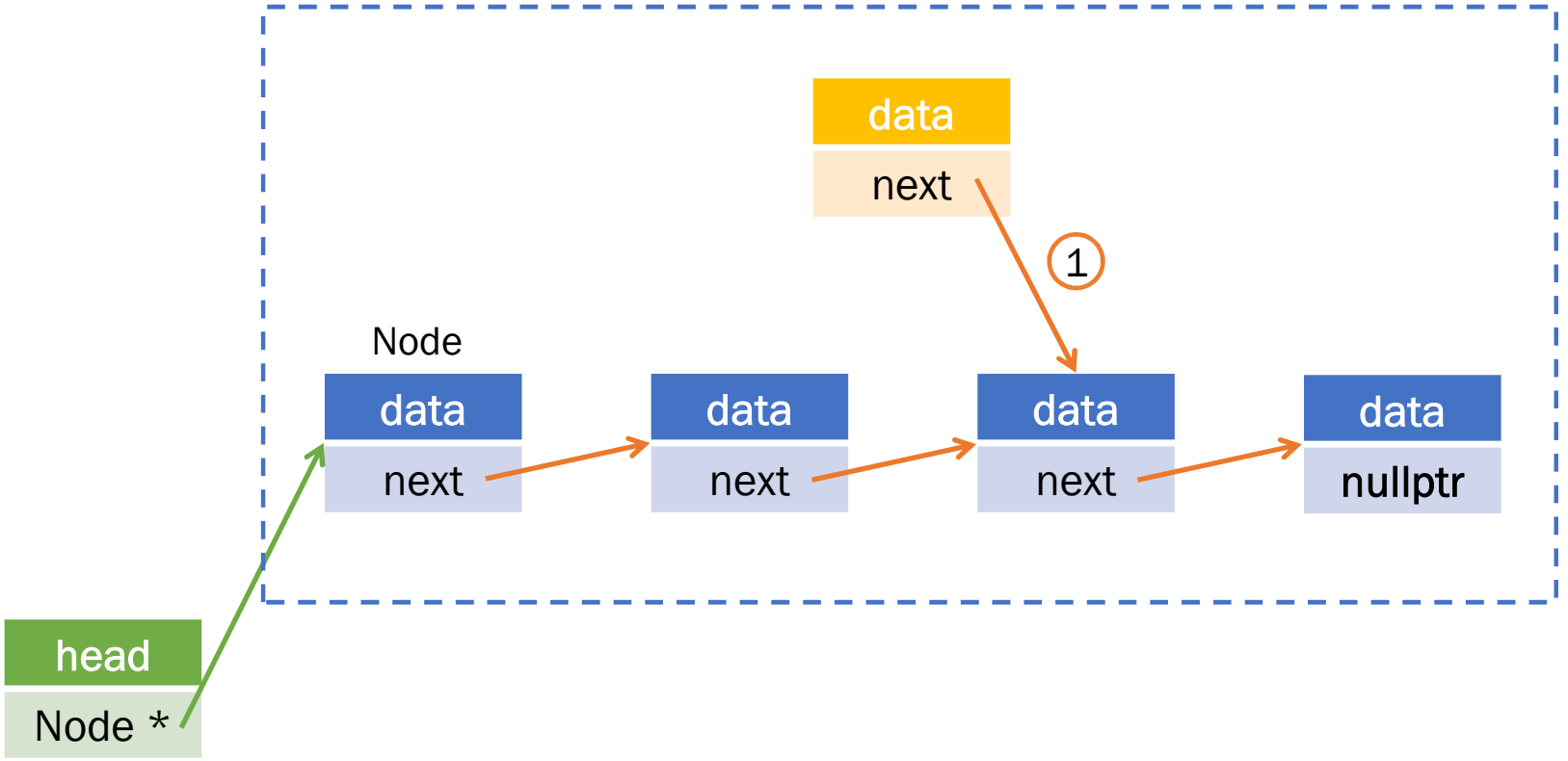
Односвязный список. Вставка узла



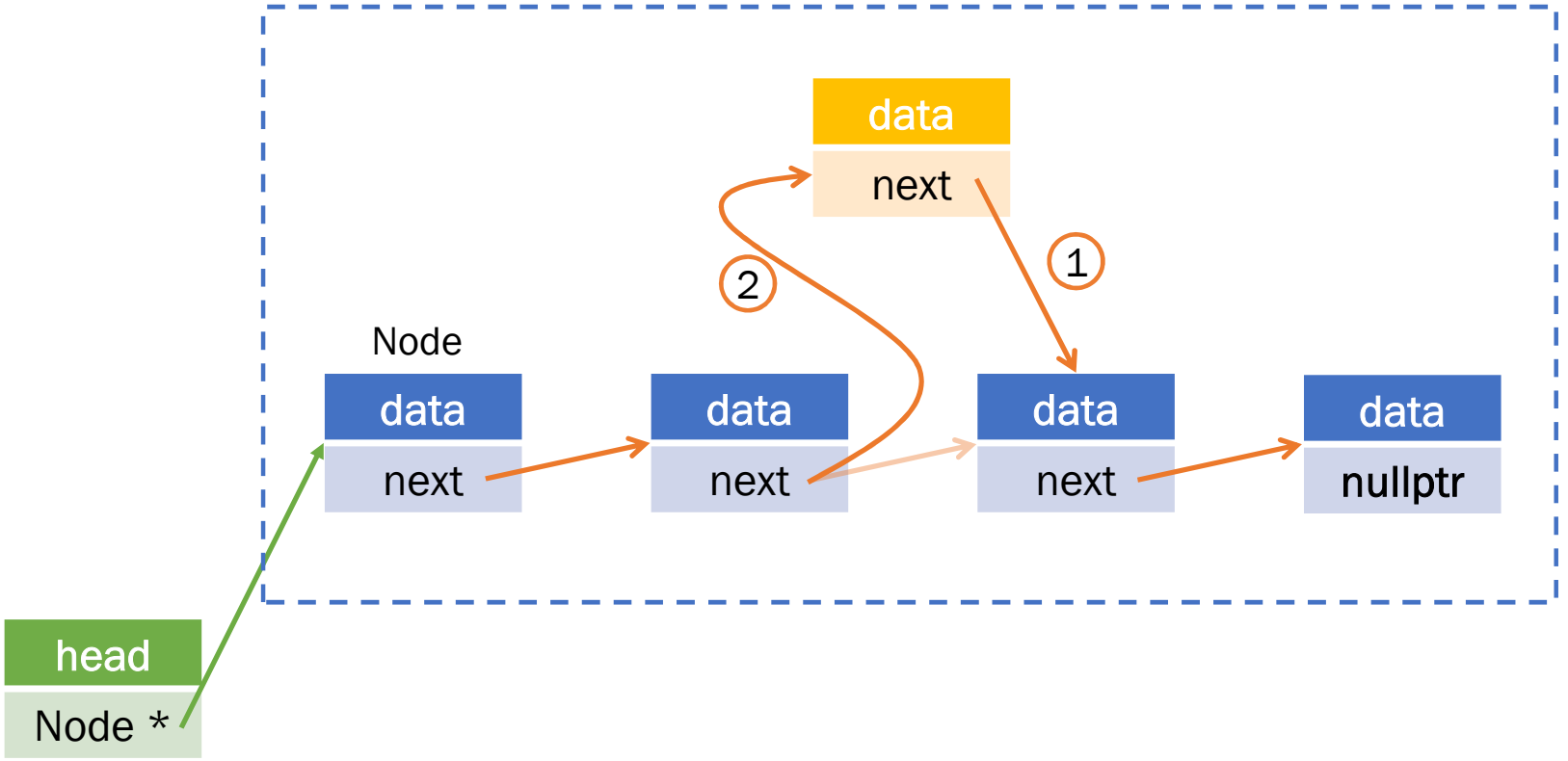
Односвязный список. Вставка узла



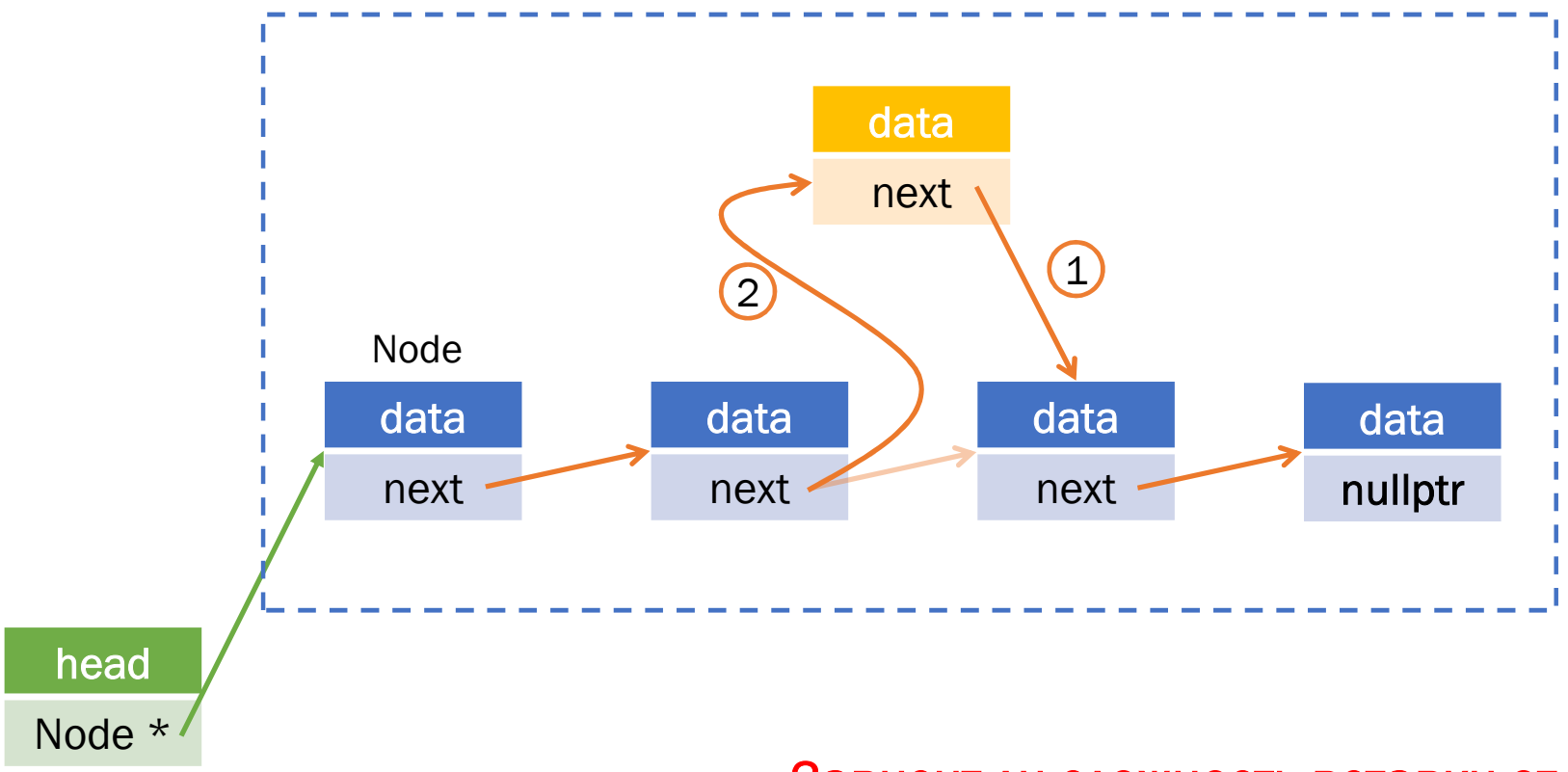
Односвязный список. Вставка узла



Односвязный список. Вставка узла

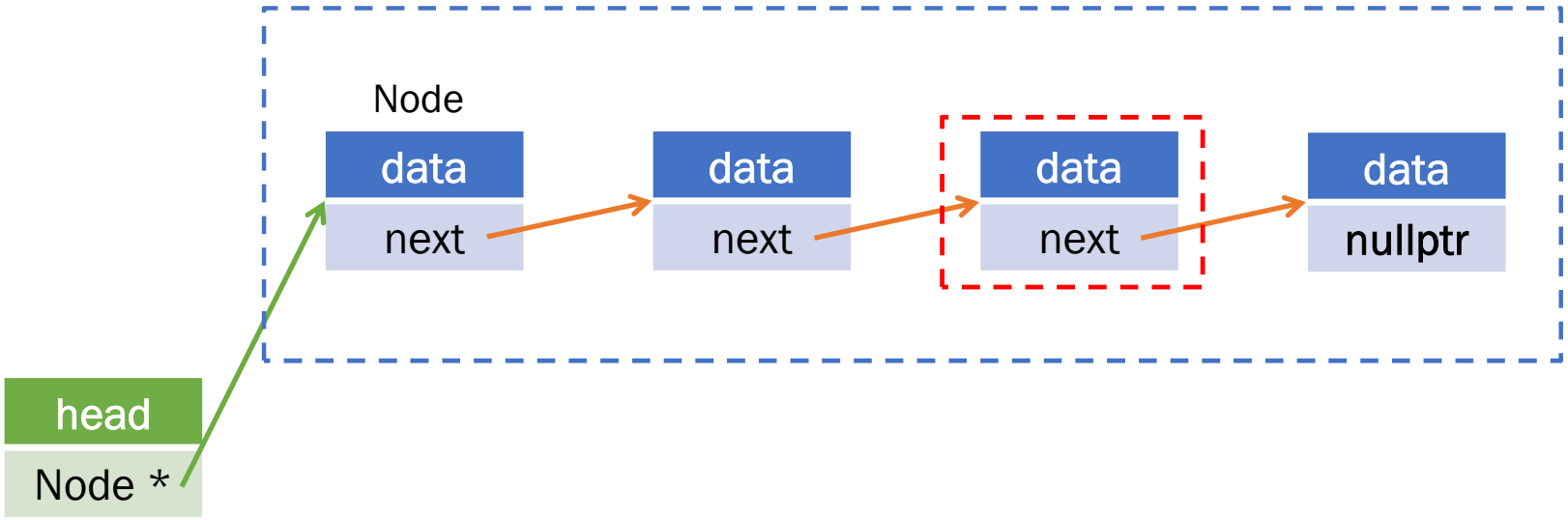


Односвязный список. Вставка узла

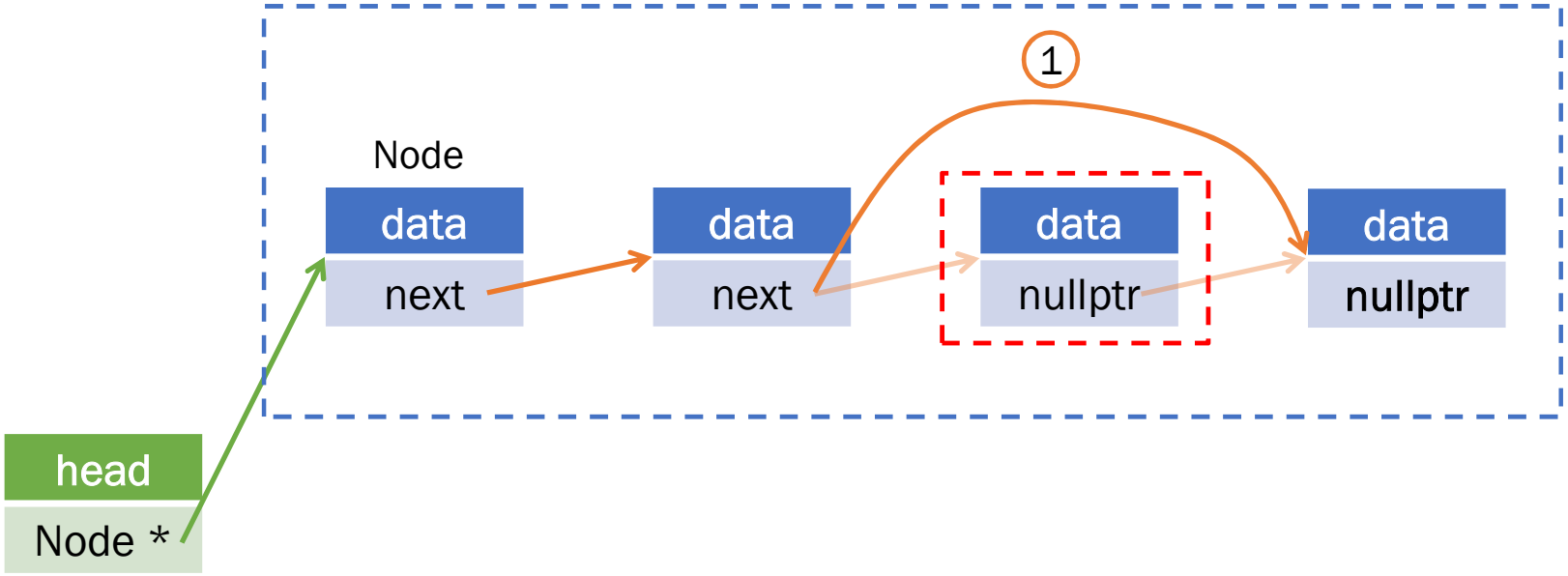


Зависит ли сложность вставки от длины списка?

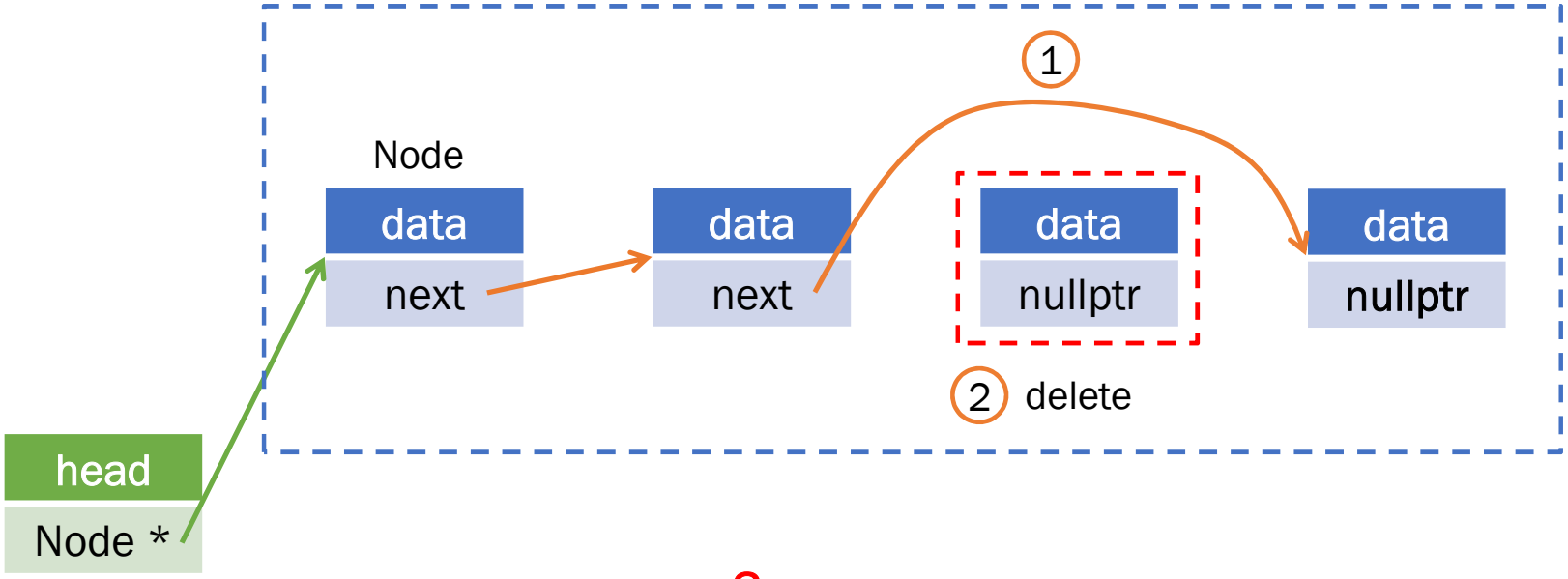
Односвязный список. Удаление узла



Односвязный список. Удаление узла



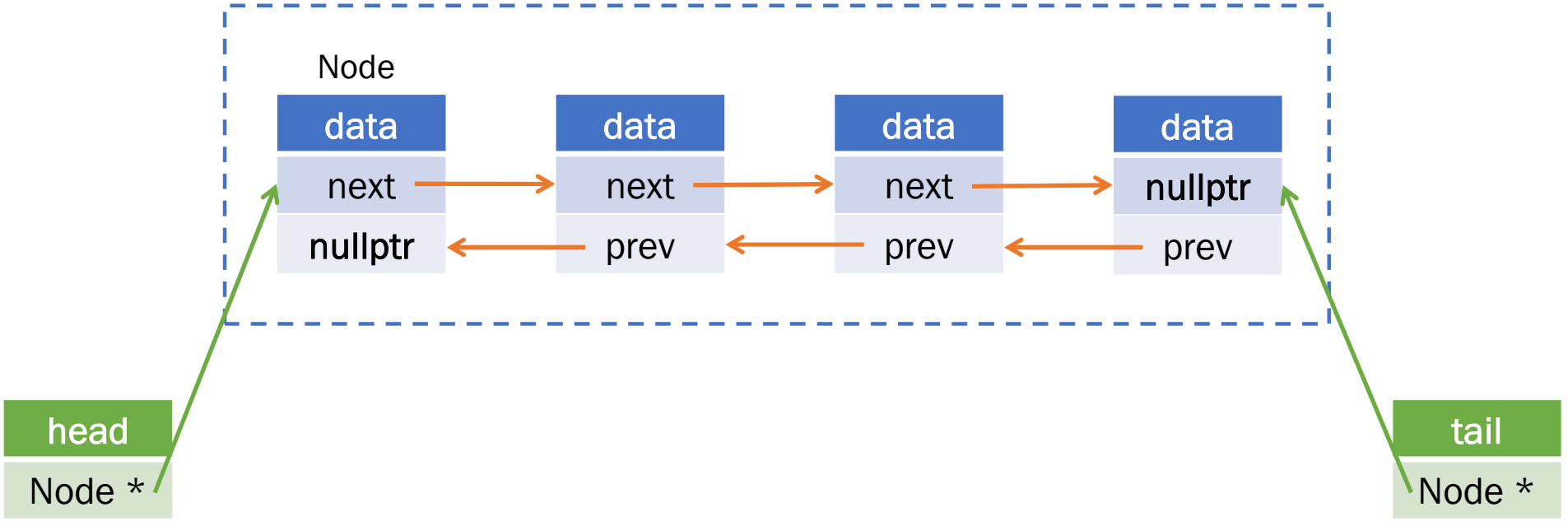
Односвязный список. Удаление узла



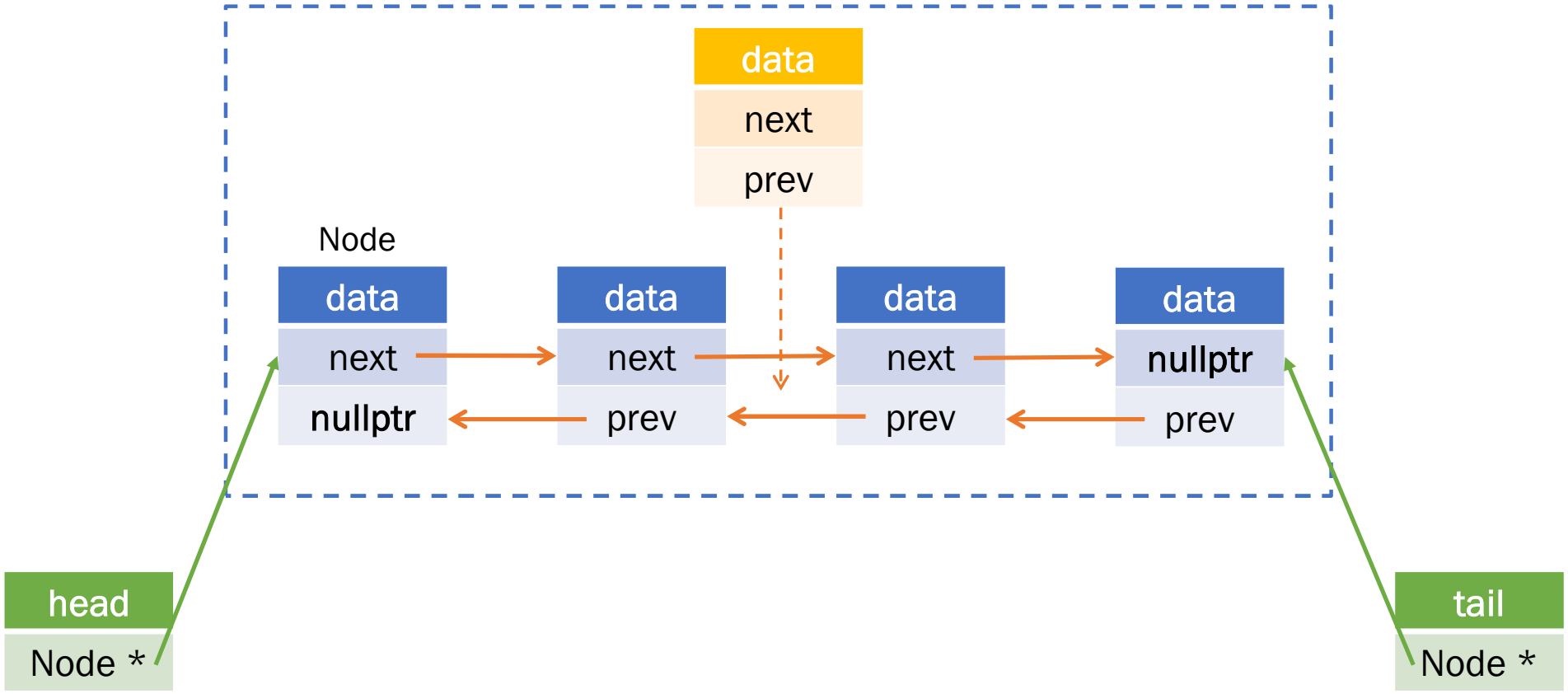
Зависит ли сложность удаления от длины списка?

Двусвязный список

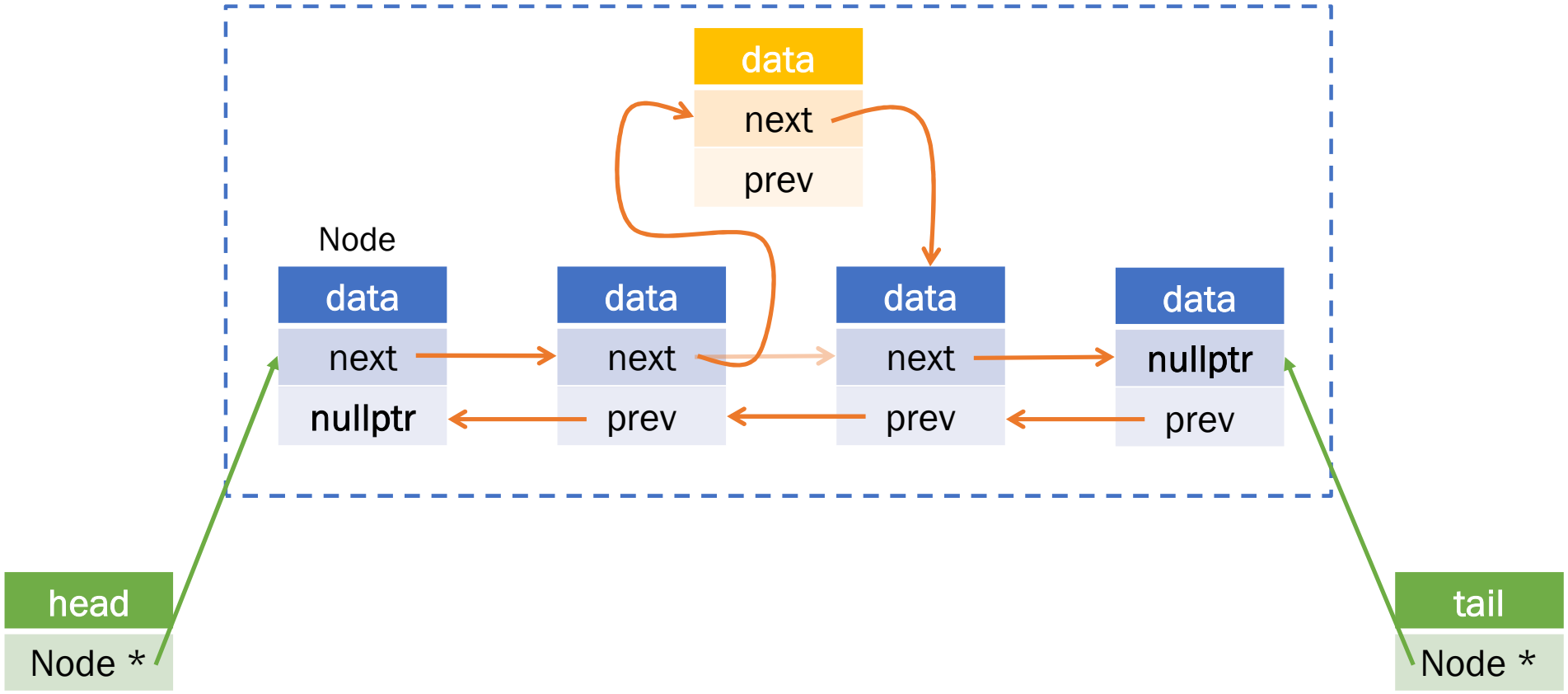
Двусвязный список. Вставка узла



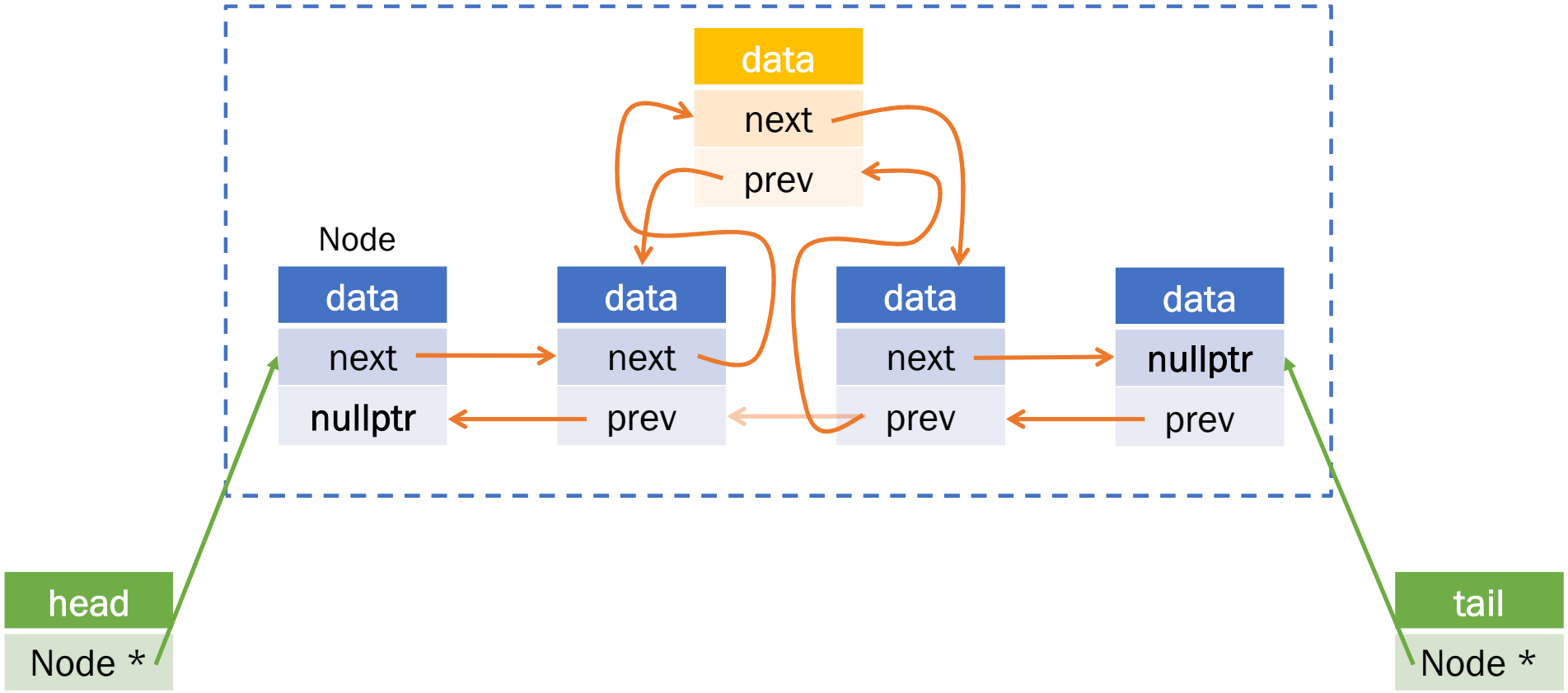
Двусвязный список. Вставка узла



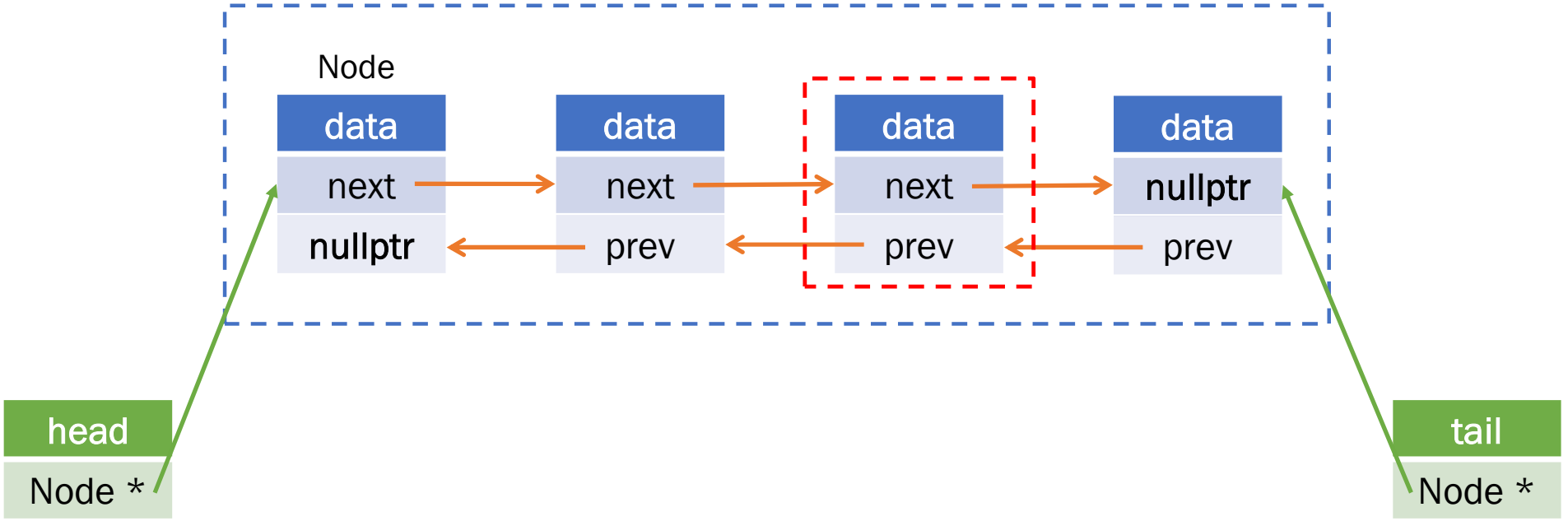
Двусвязный список. Вставка узла



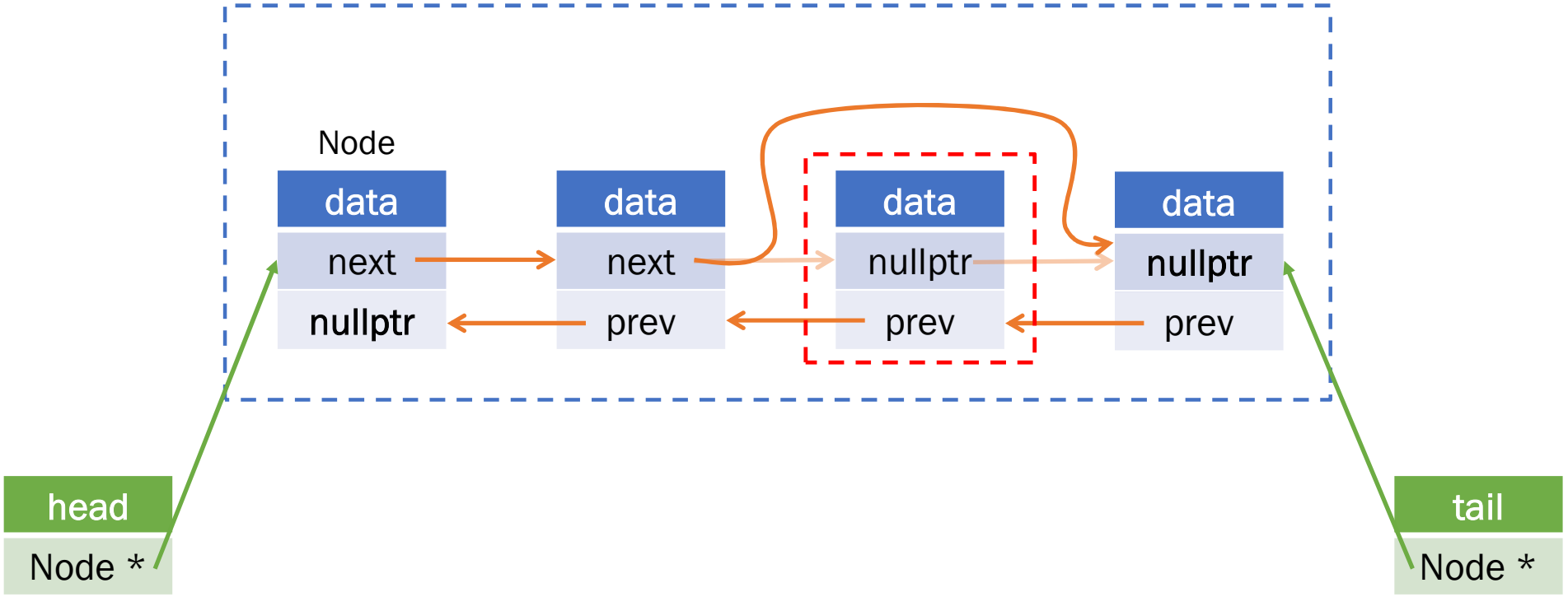
Двусвязный список. Вставка узла



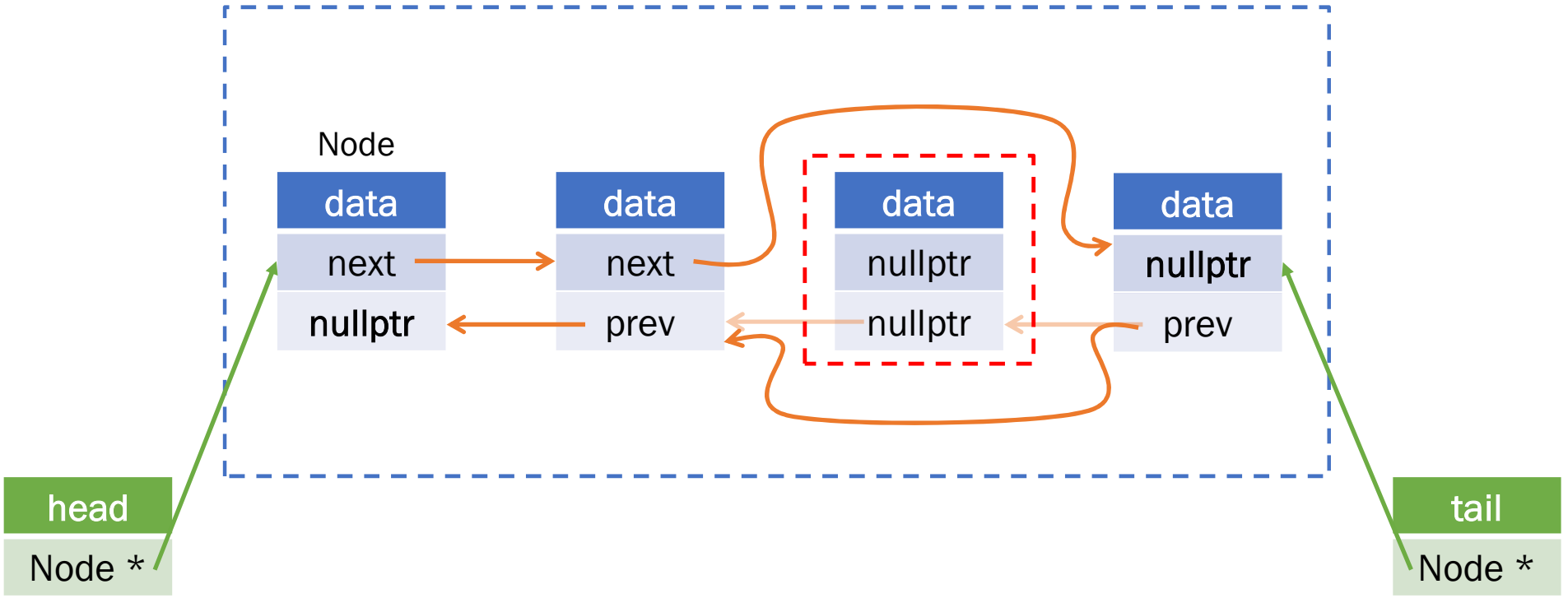
Двусвязный список. Удаление узла



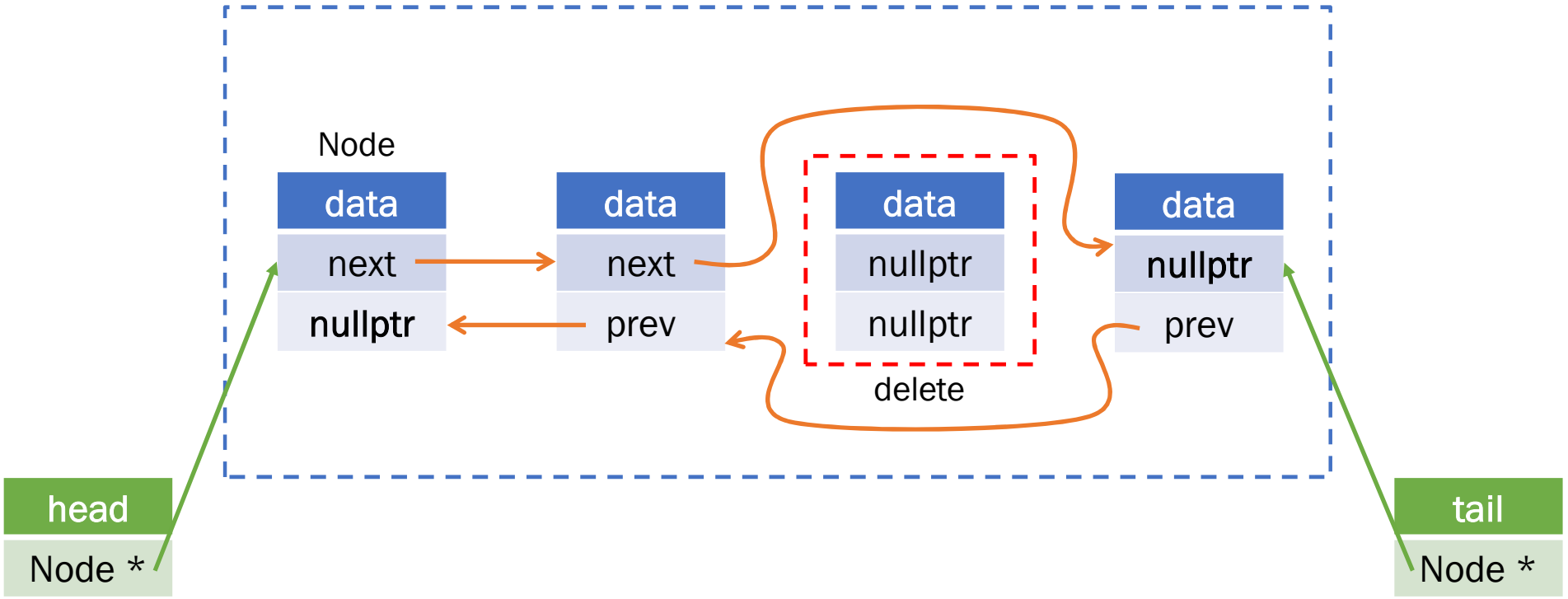
Двусвязный список. Удаление узла



Двусвязный список. Удаление узла



Двусвязный список. Удаление узла



Двусвязный список

- Двусвязный список можно обходить в обоих направлениях (`head->tail`, `tail->head`)
- Удаление элемента более эффективно, так как хранится указатель на предыдущий узел списка (`prev`)

Двусвязный список

- Двусвязный список можно обходить в обоих направлениях (`head->tail`, `tail->head`)
- Удаление элемента более эффективно, так как хранится указатель на предыдущий узел списка (`prev`)
- Необходимость хранения дополнительного указателя
- Выполнение операций связано с дополнительными модификациями указателей

XOR-связный список

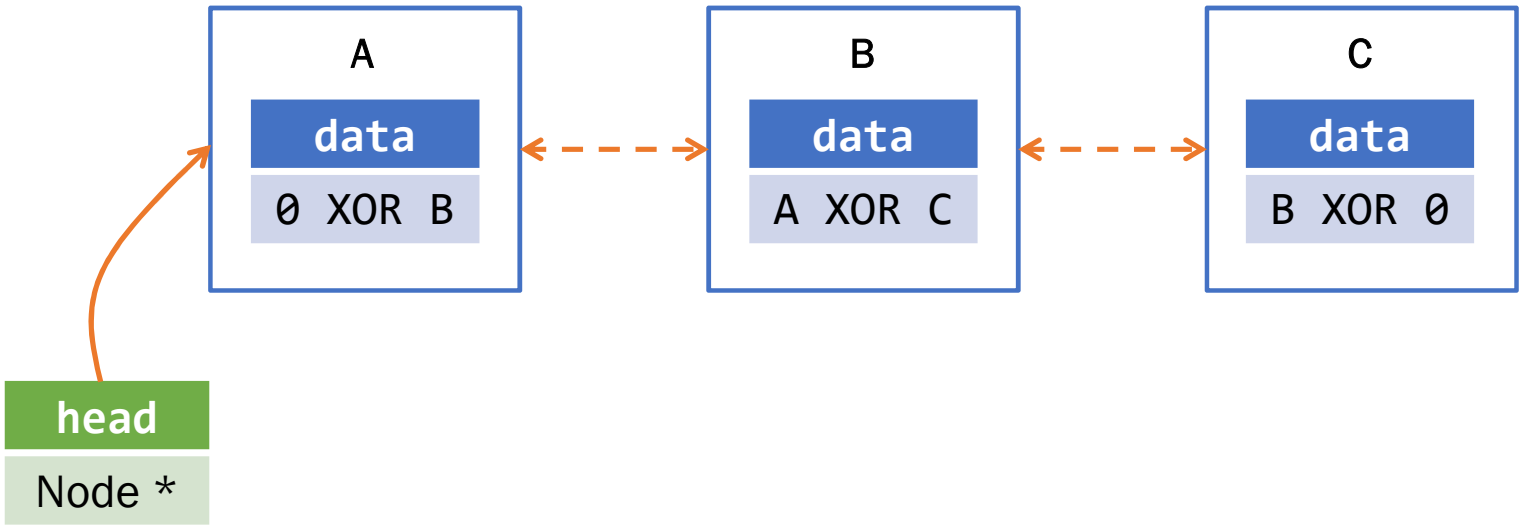
XOR-связный список

Исключающее или – XOR (^)

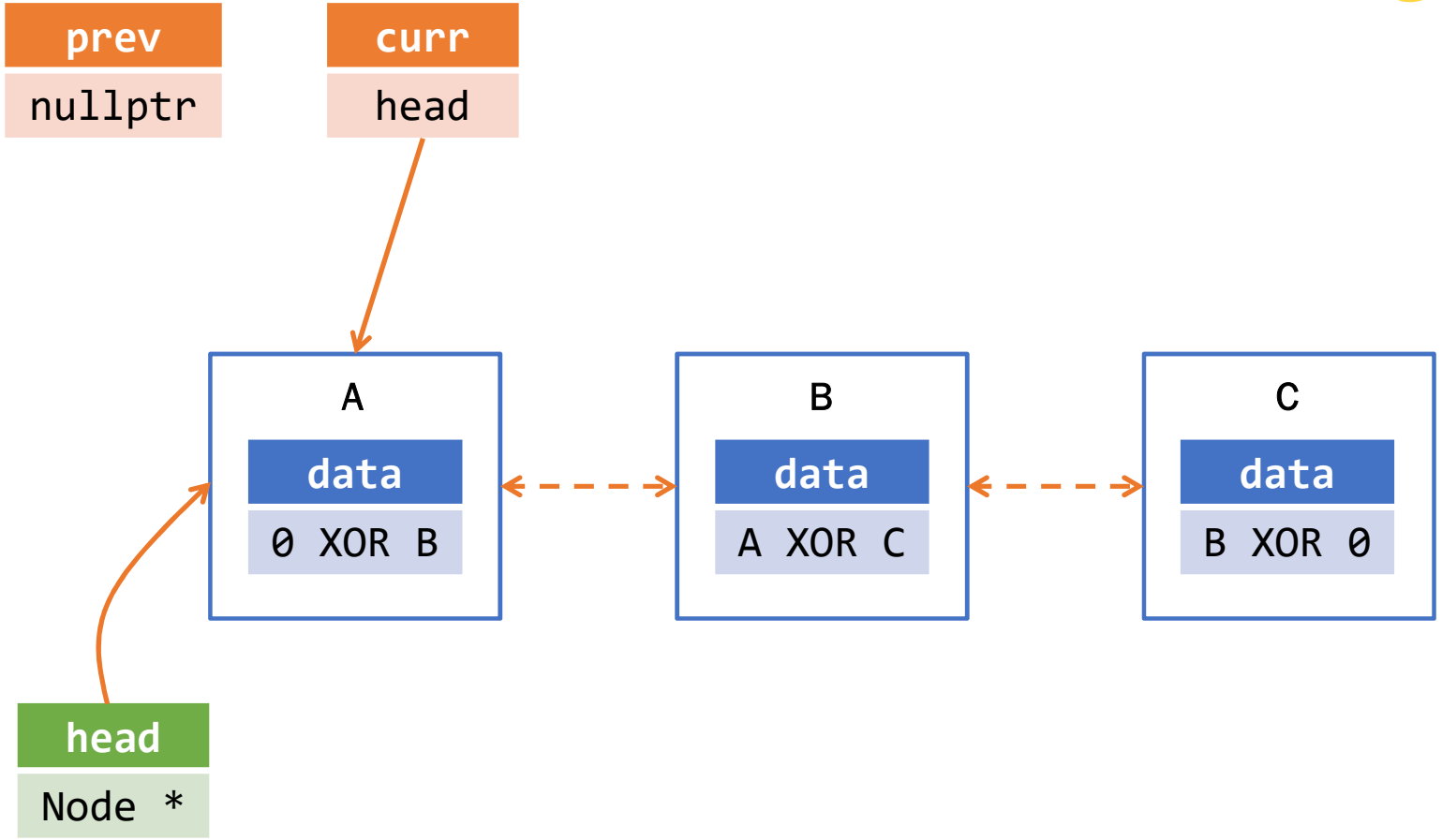
- A XOR 0 = A
- A XOR A = 0
- A XOR B = B XOR A
- A XOR (B XOR C) = (A XOR B) XOR C
- (A XOR B) XOR A = B

XOR-связный список

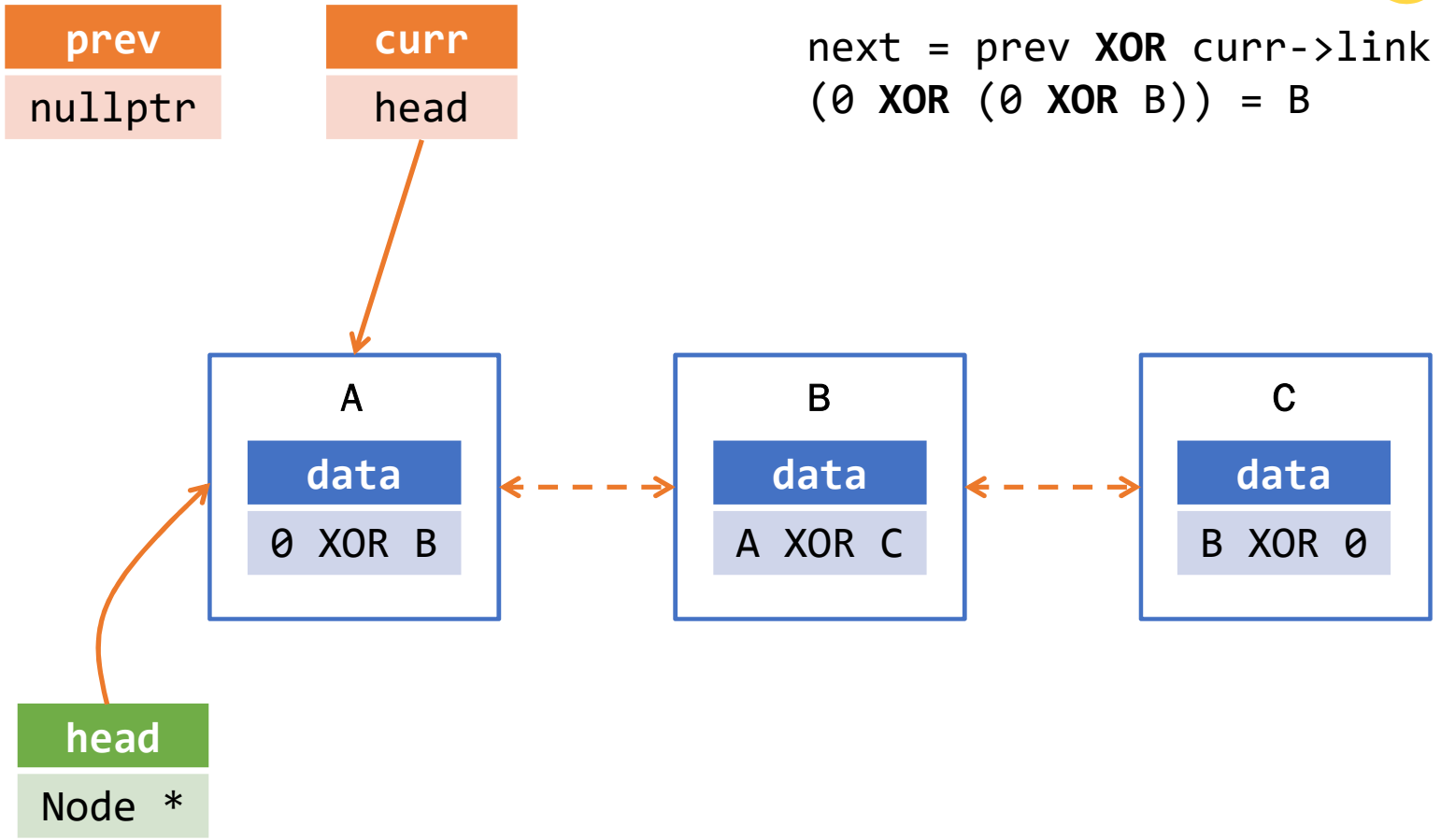
Вместо хранения обоих адресов, храним XOR адресов предыдущего и следующего узлов списка.



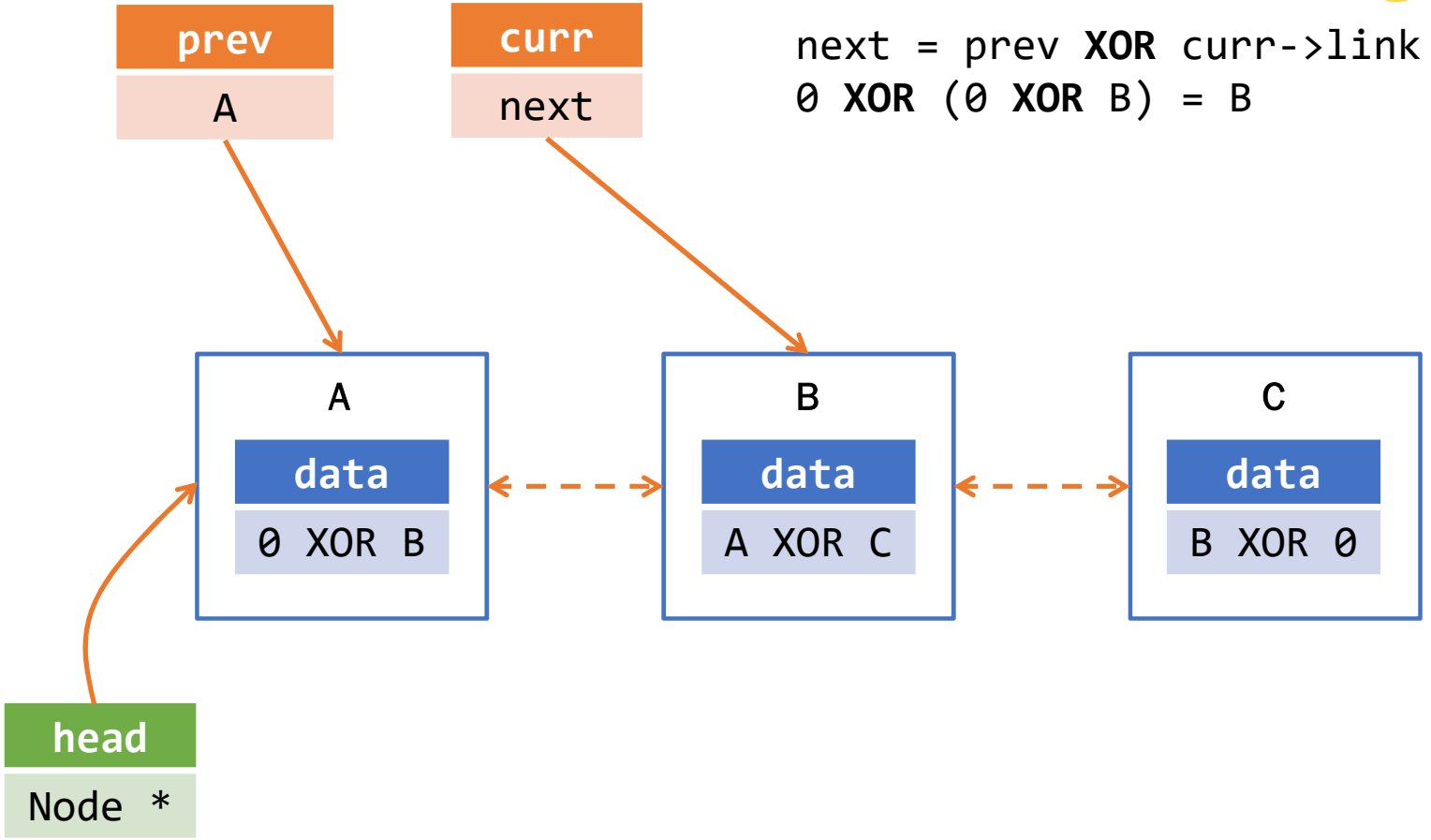
XOR-связный список. Обход



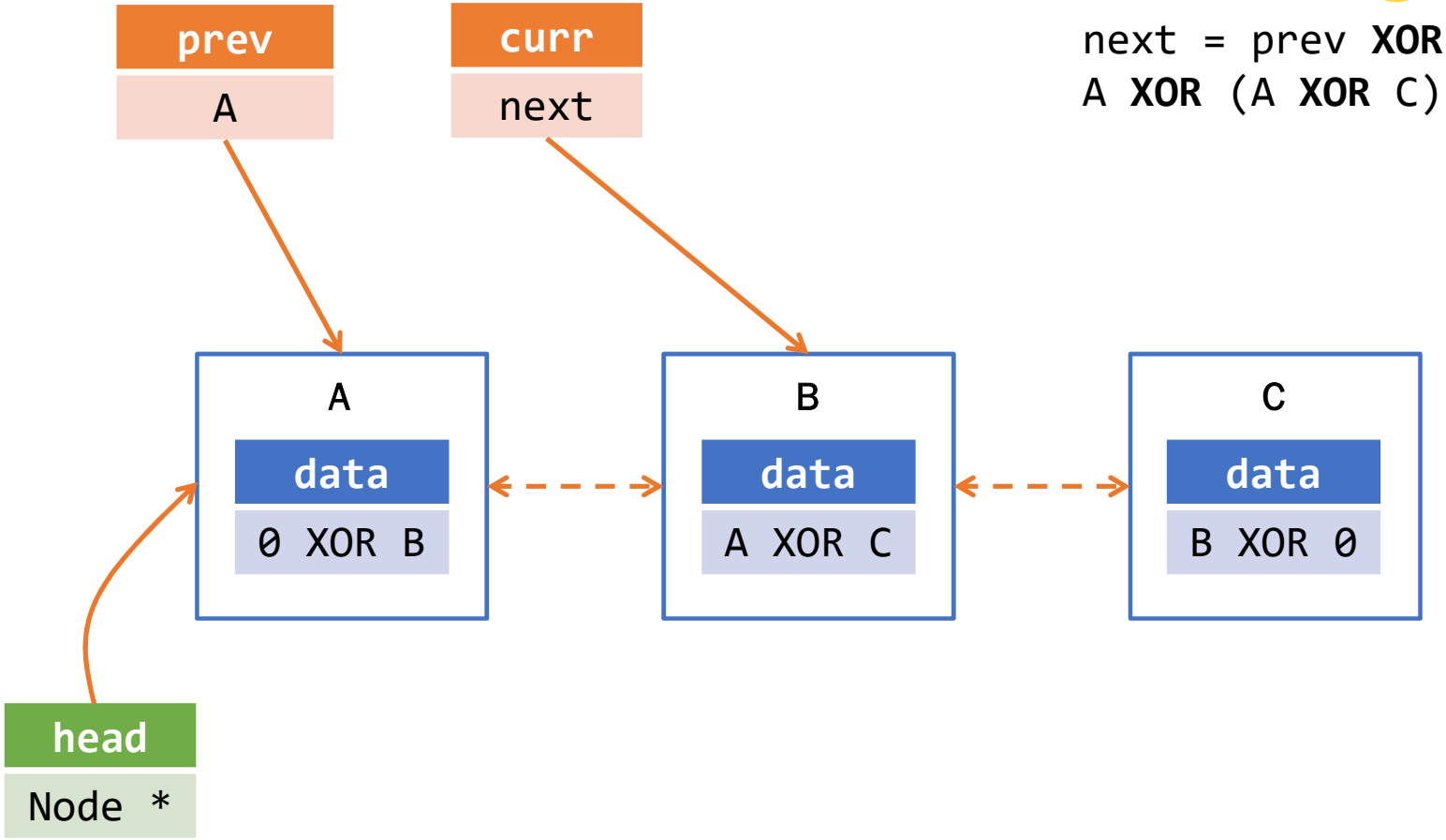
XOR-связный список. Обход



XOR-связный список. Обход

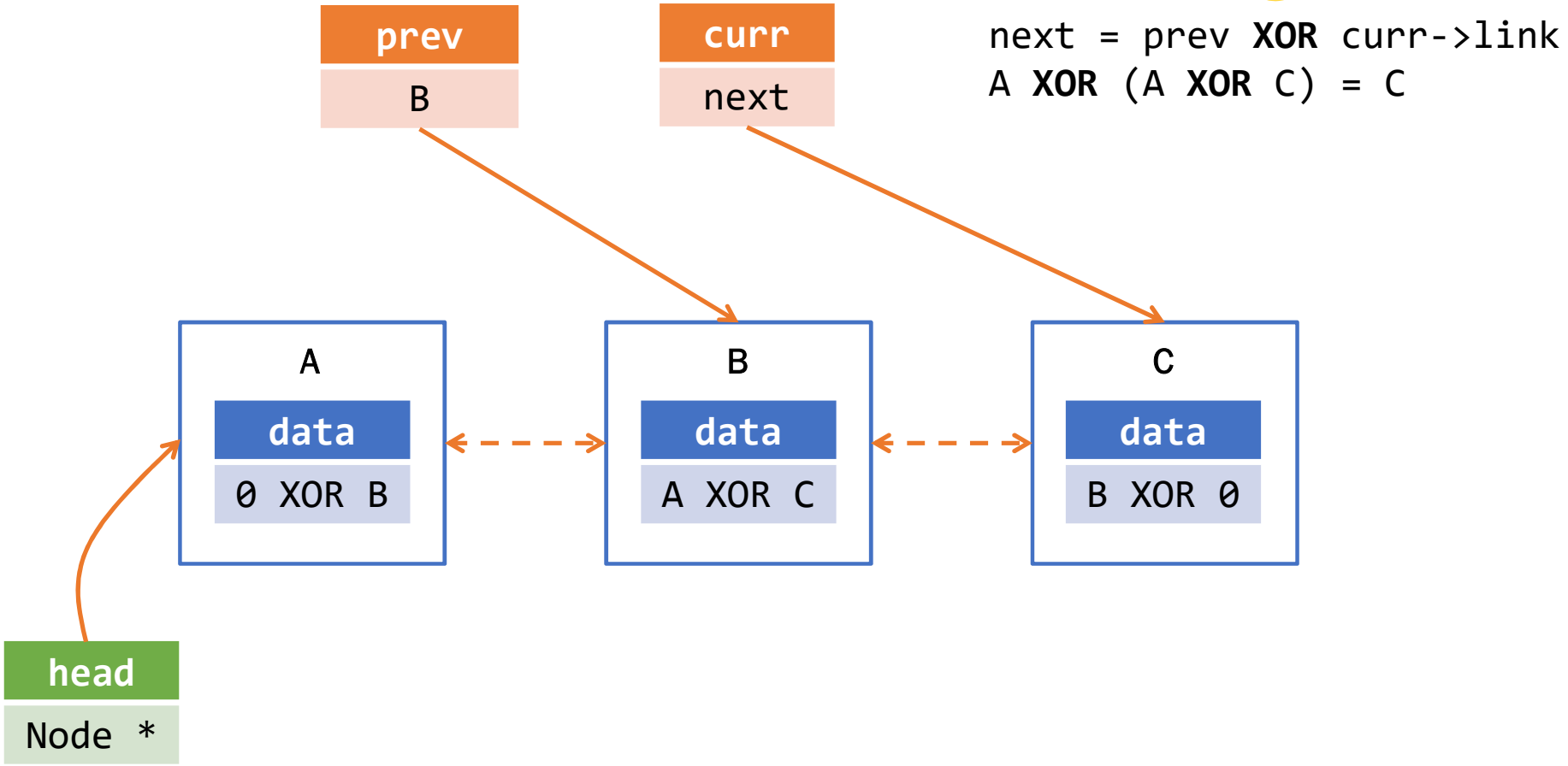


XOR-связный список. Обход




```
next = prev XOR curr->link
A XOR (A XOR C) = C
```

XOR-связный список. Обход



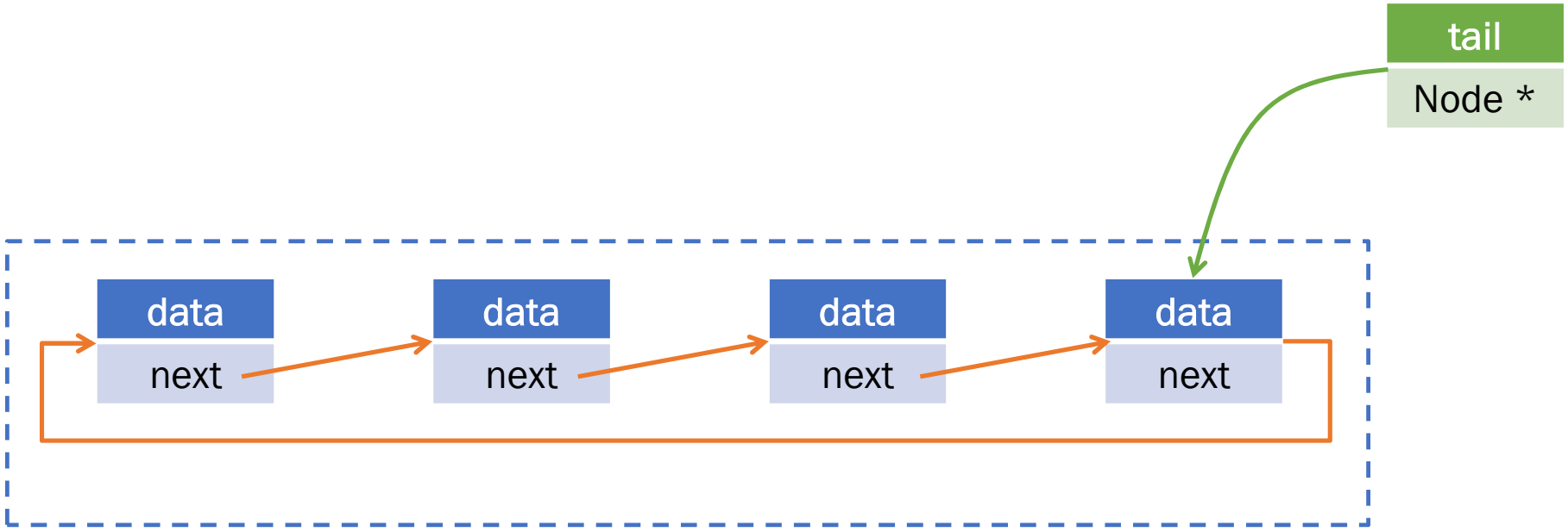
XOR-связный список. Недостатки

- Двусвязный список легче в реализации и поддержке
- Операция XOR над указателями поддерживается не всеми языками и требует использования специфических средств
- Иметь указатель на одну вершину (в середине такого списка) недостаточно для удаления или вставки новой вершины

Циклический список

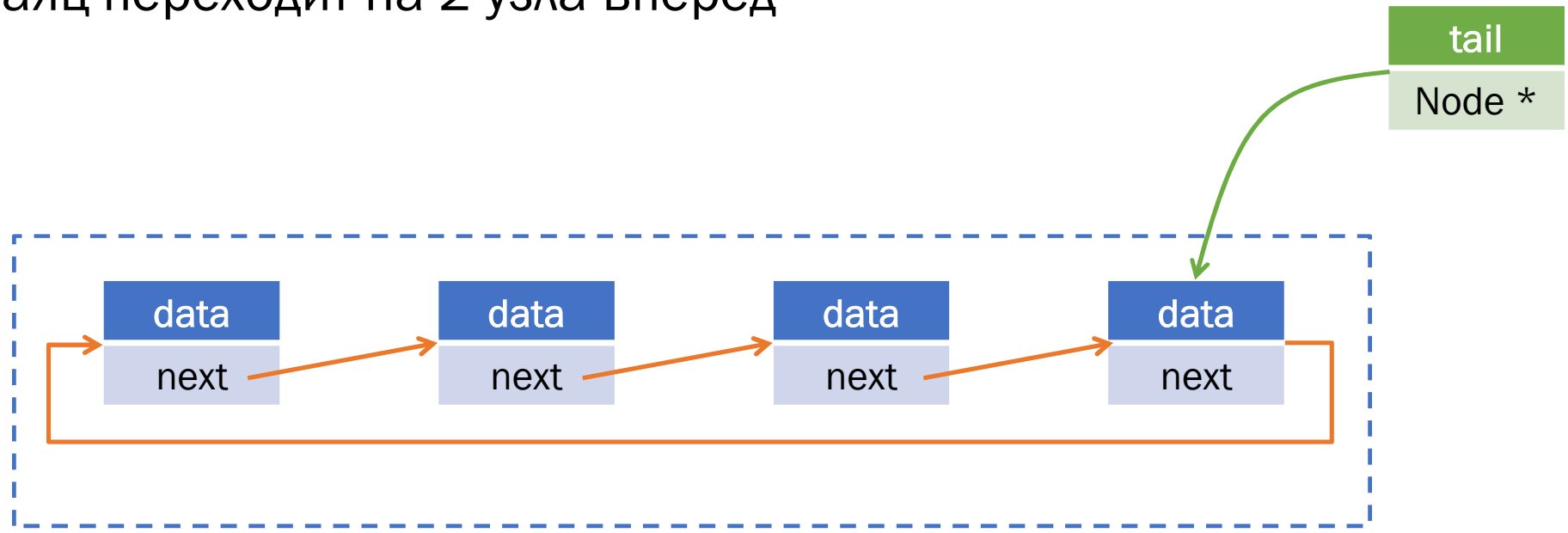
Циклический односвязный список

Первый узел списка является следующим для последнего.



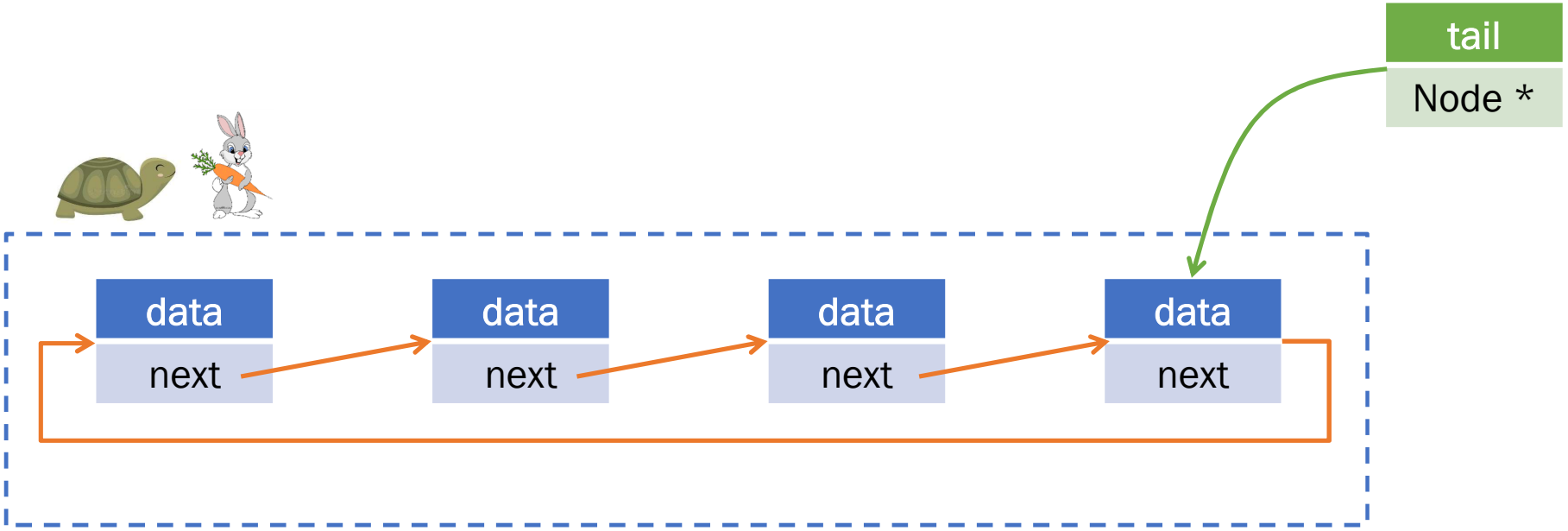
Поиск цикла в списке. Алгоритм Флойда

- Черепаха переходит на 1 узел вперед
- Заяц переходит на 2 узла вперед



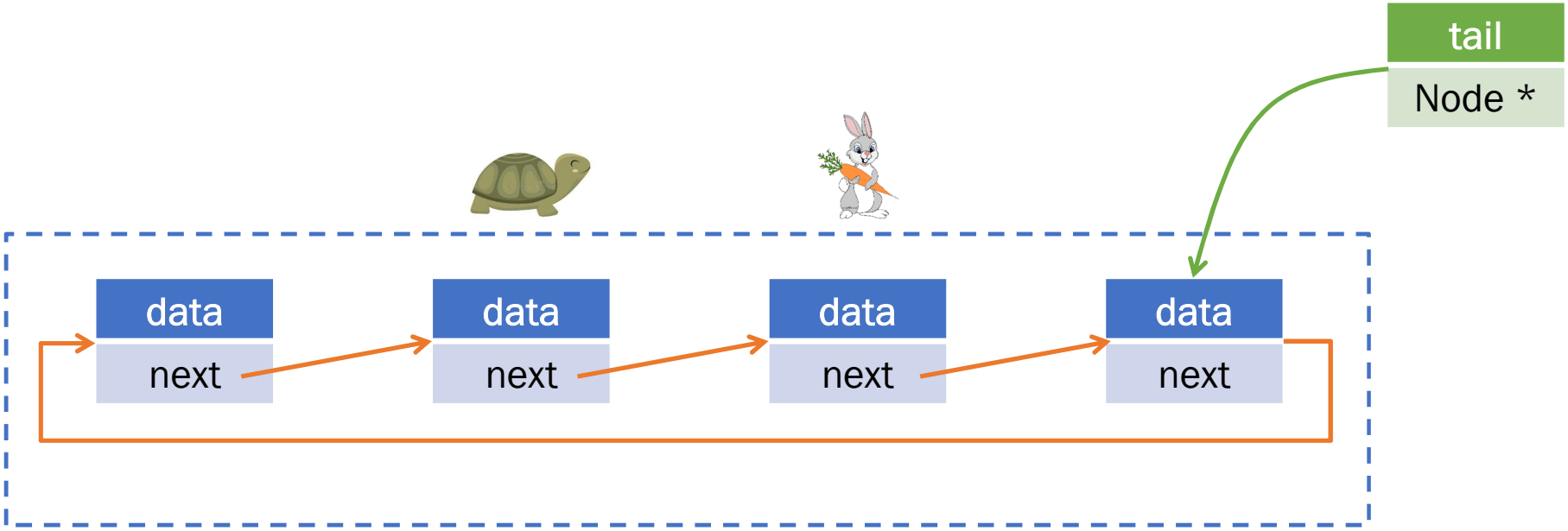
Поиск цикла в списке. Алгоритм Флойда

- Черепаха переходит на 1 узел вперед
- Заяц переходит на 2 узла вперед



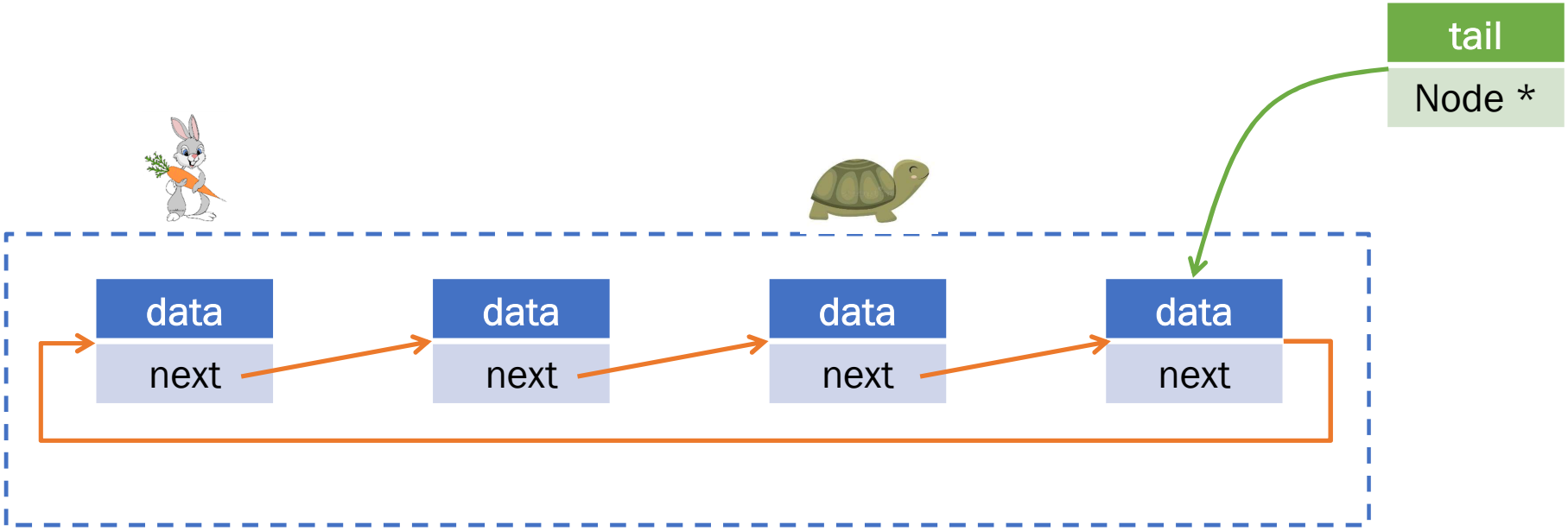
Поиск цикла в списке. Алгоритм Флойда

- Черепаха переходит на 1 узел вперед
- Заяц переходит на 2 узла вперед



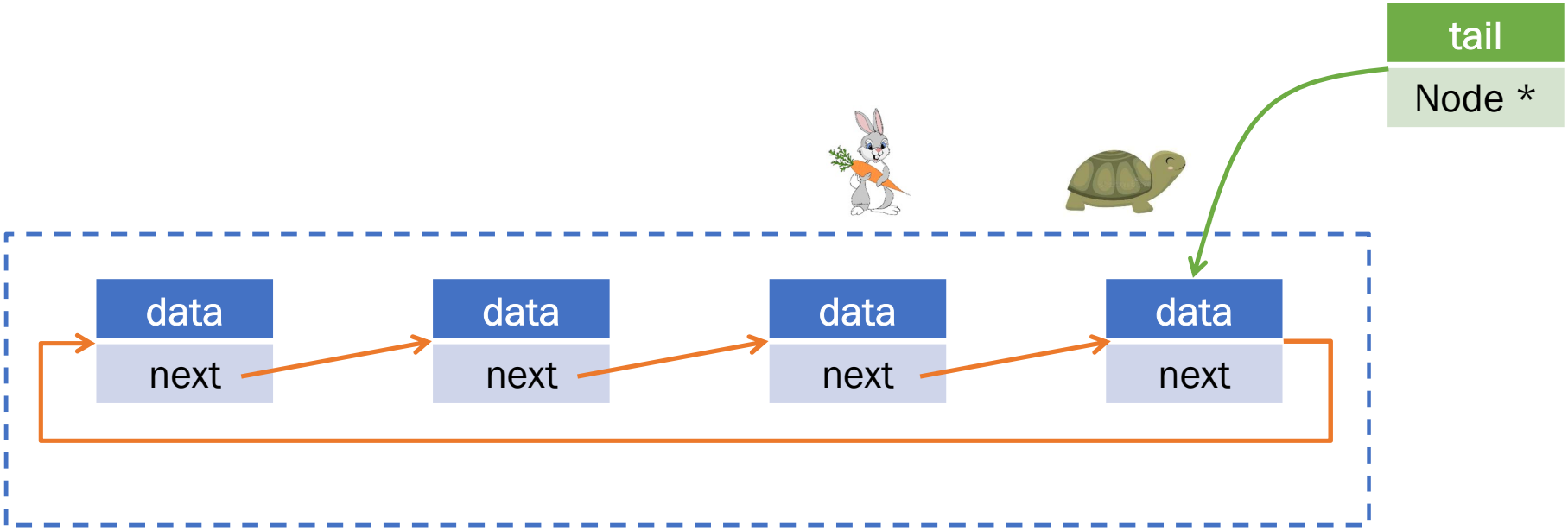
Поиск цикла в списке. Алгоритм Флойда

- Черепаха переходит на 1 узел вперед
- Заяц переходит на 2 узла вперед



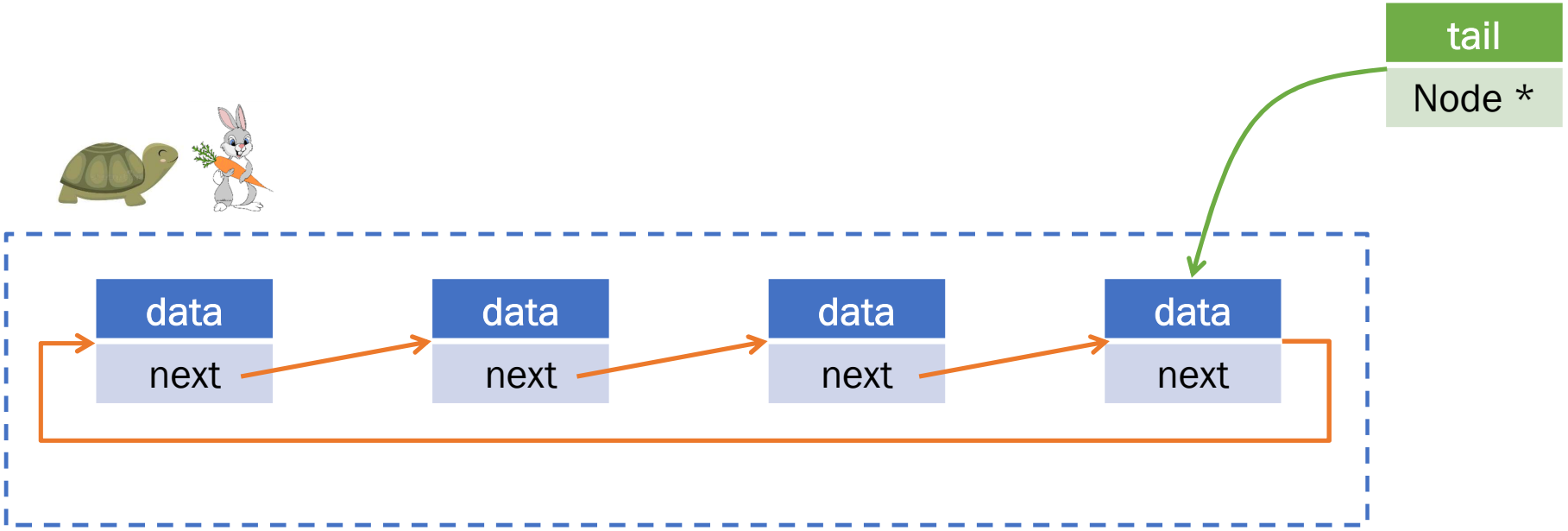
Поиск цикла в списке. Алгоритм Флойда

- Черепаха переходит на 1 узел вперед
- Заяц переходит на 2 узла вперед



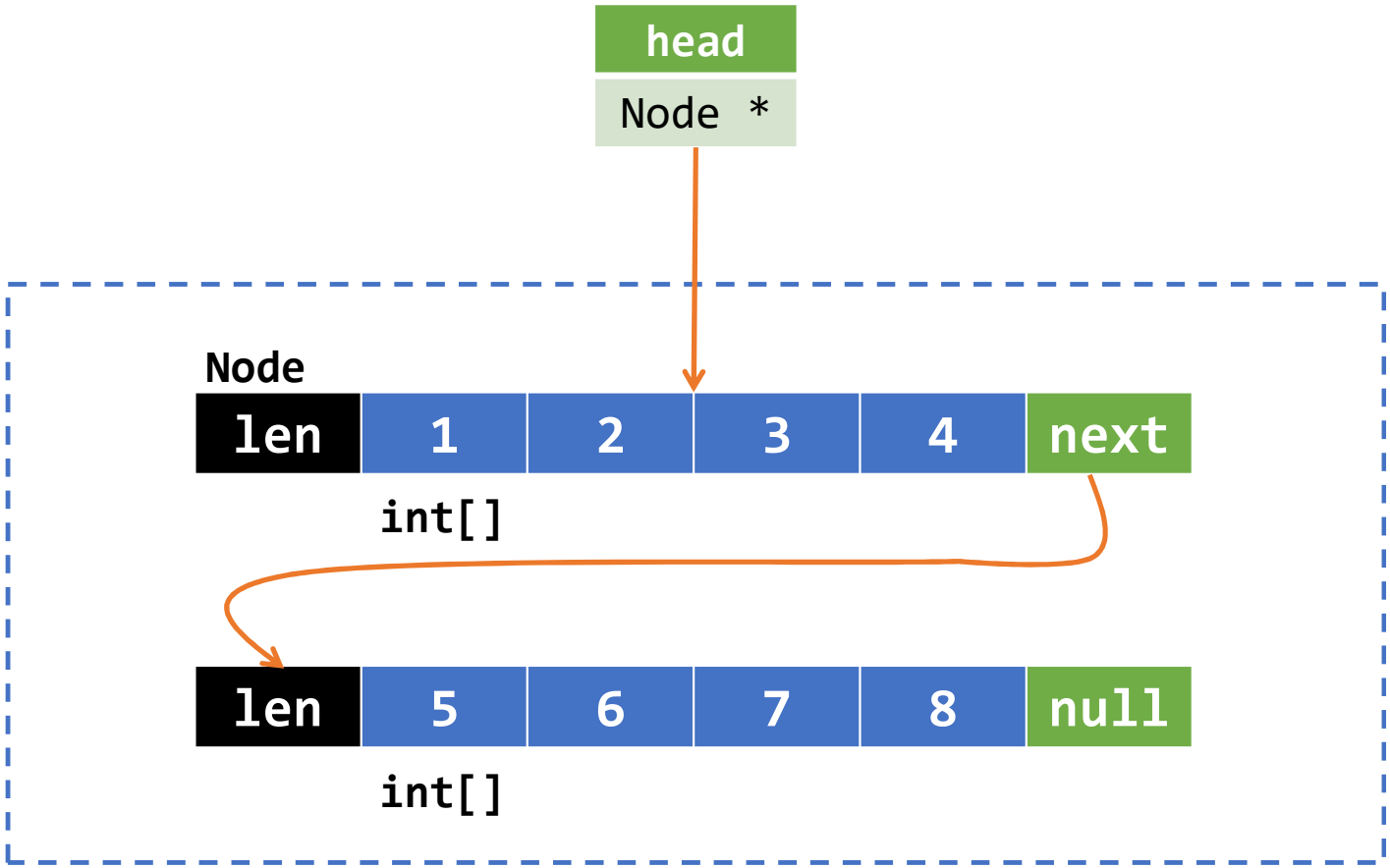
Поиск цикла в списке. Алгоритм Флойда

- Черепаха переходит на 1 узел вперед
- Заяц переходит на 2 узла вперед

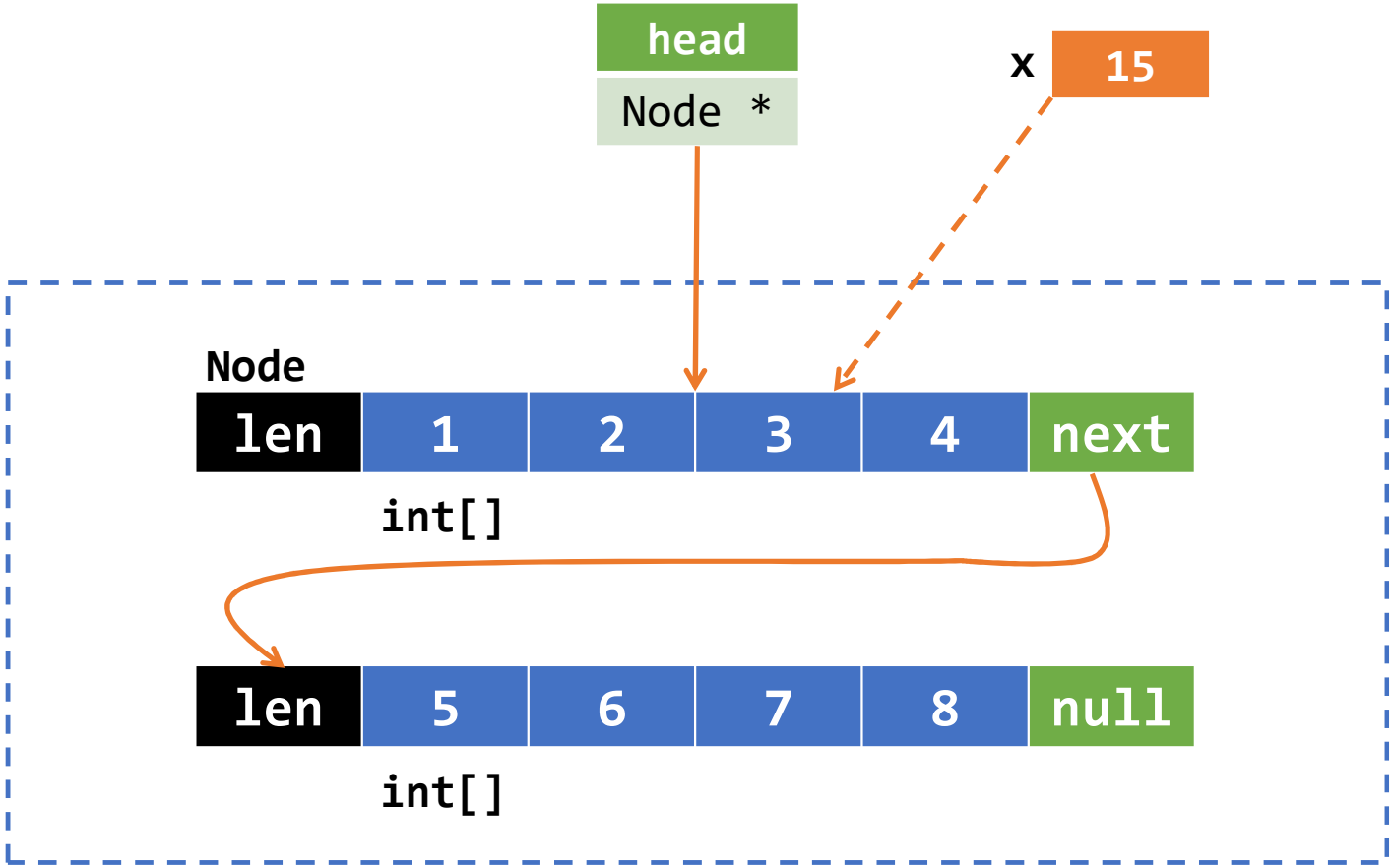


Развернутый (unrolled) список

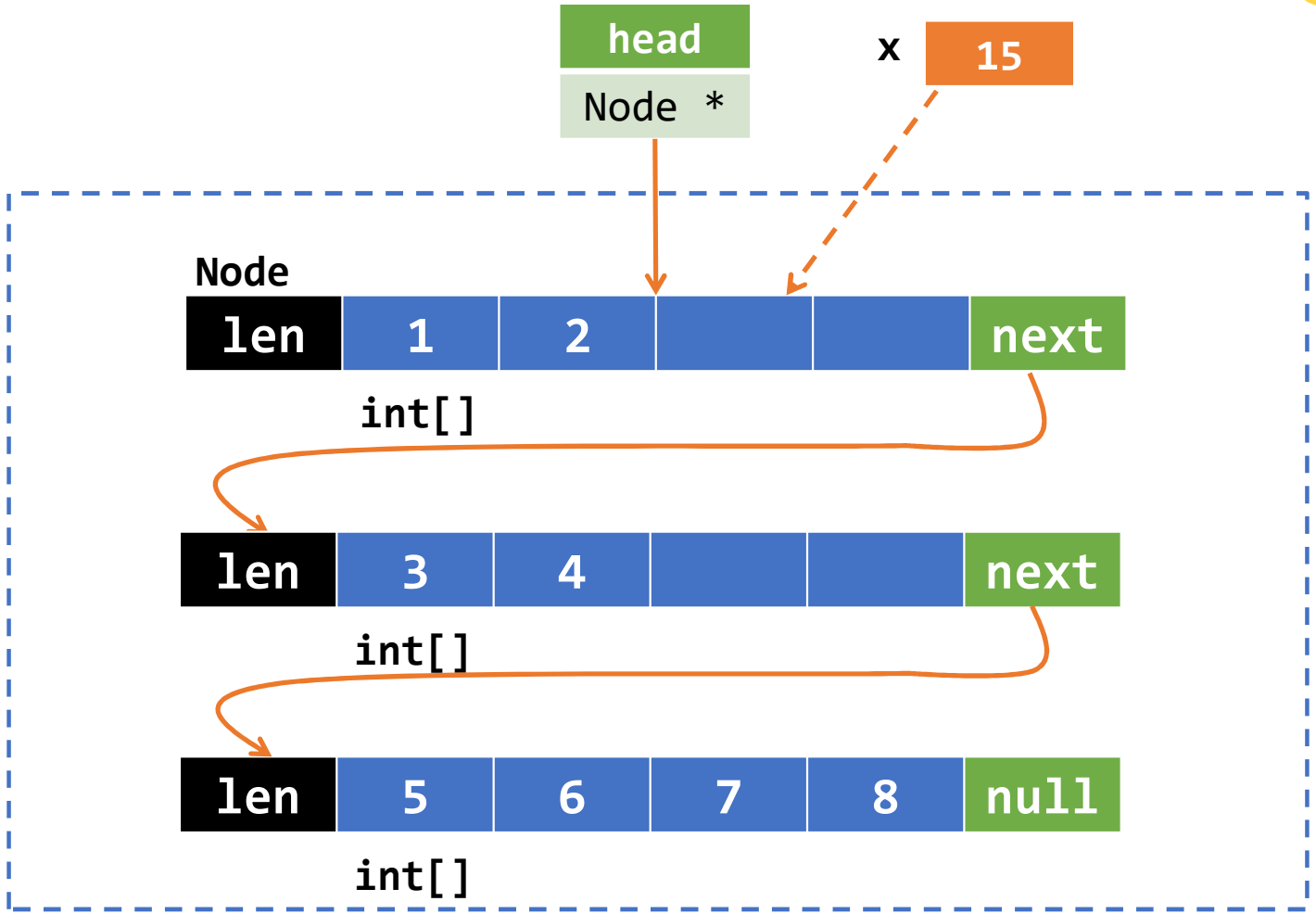
Развернутый список



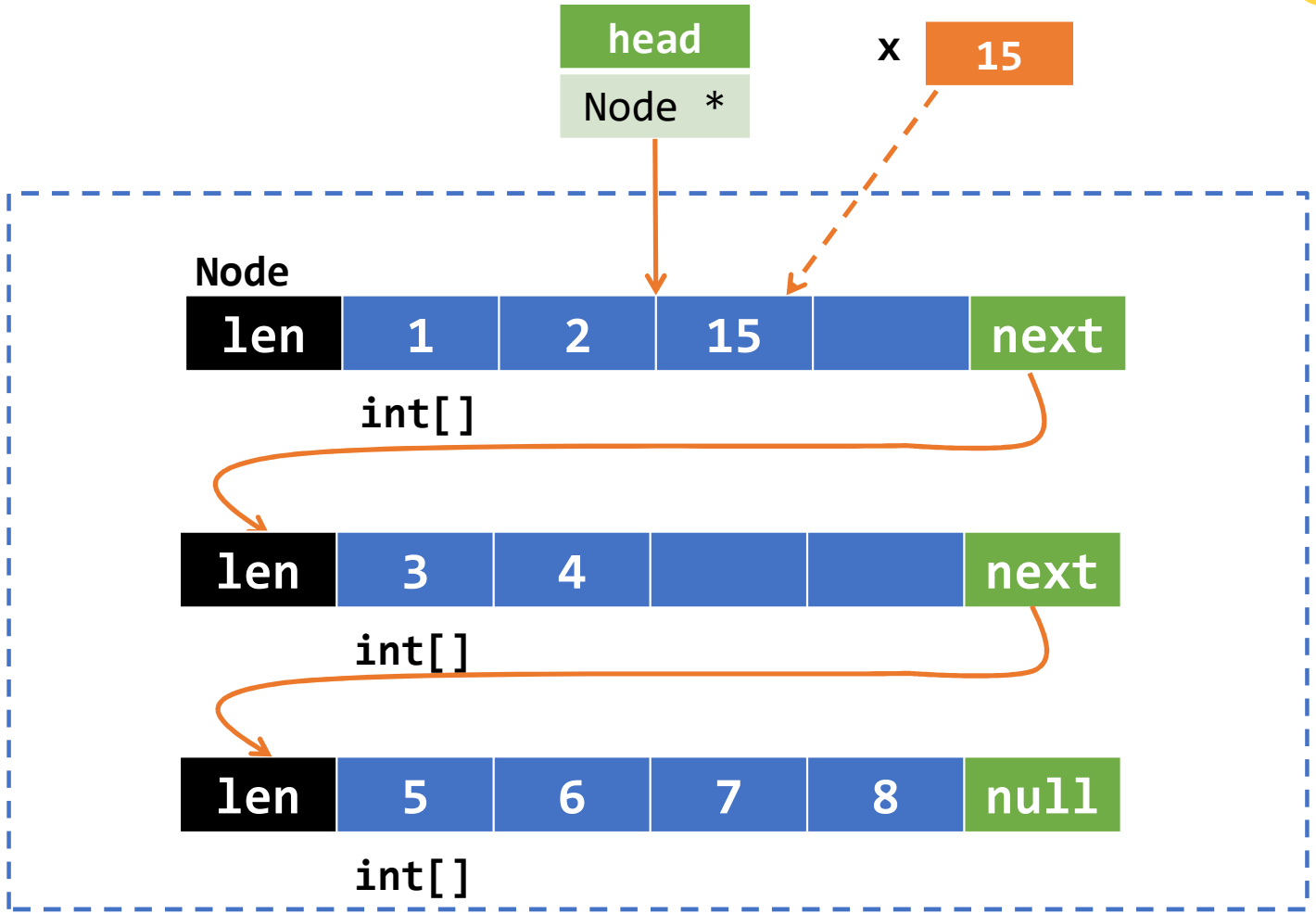
Развернутый список. Вставка значения



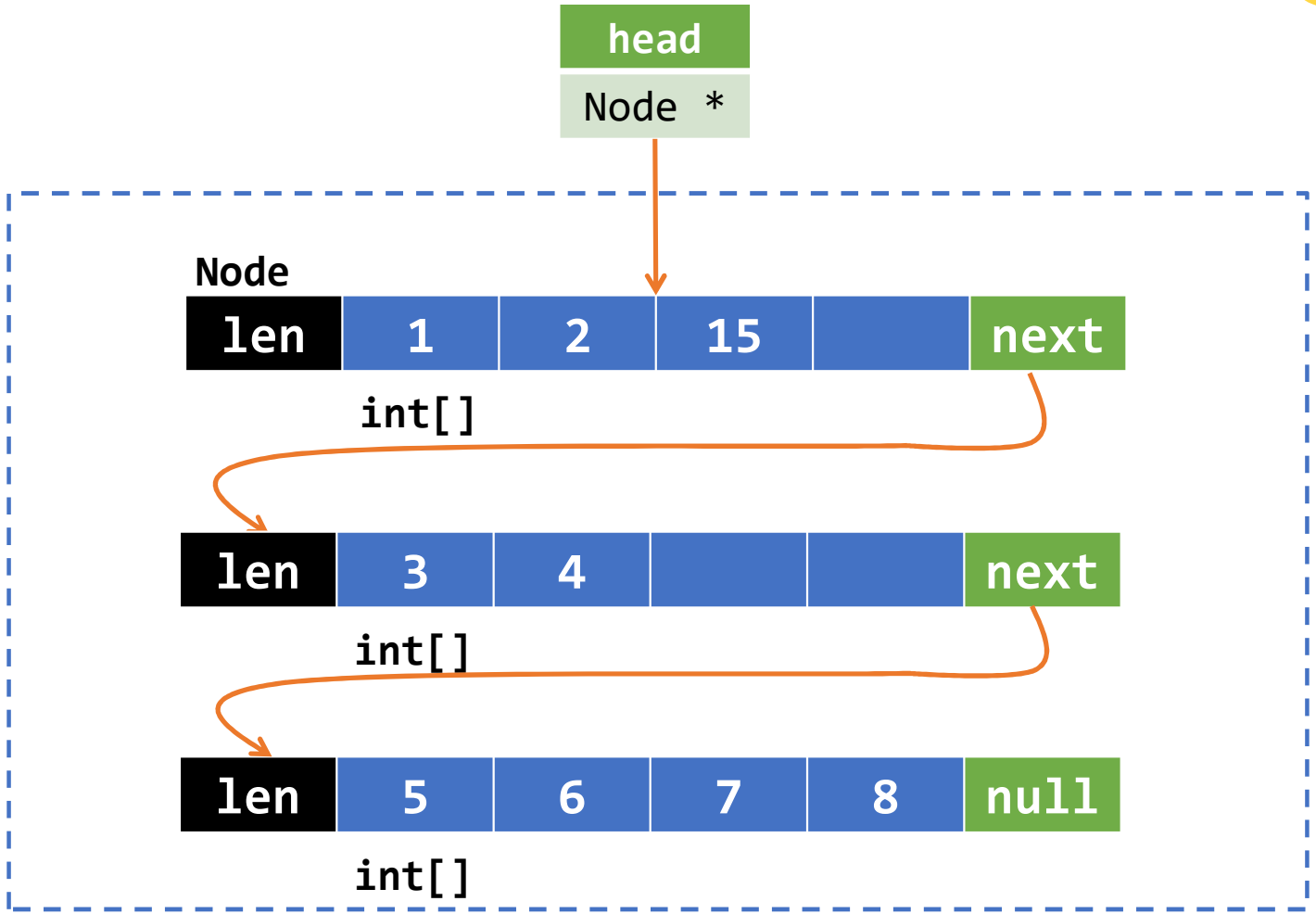
Развернутый список. Вставка значения



Развернутый список. Вставка значения

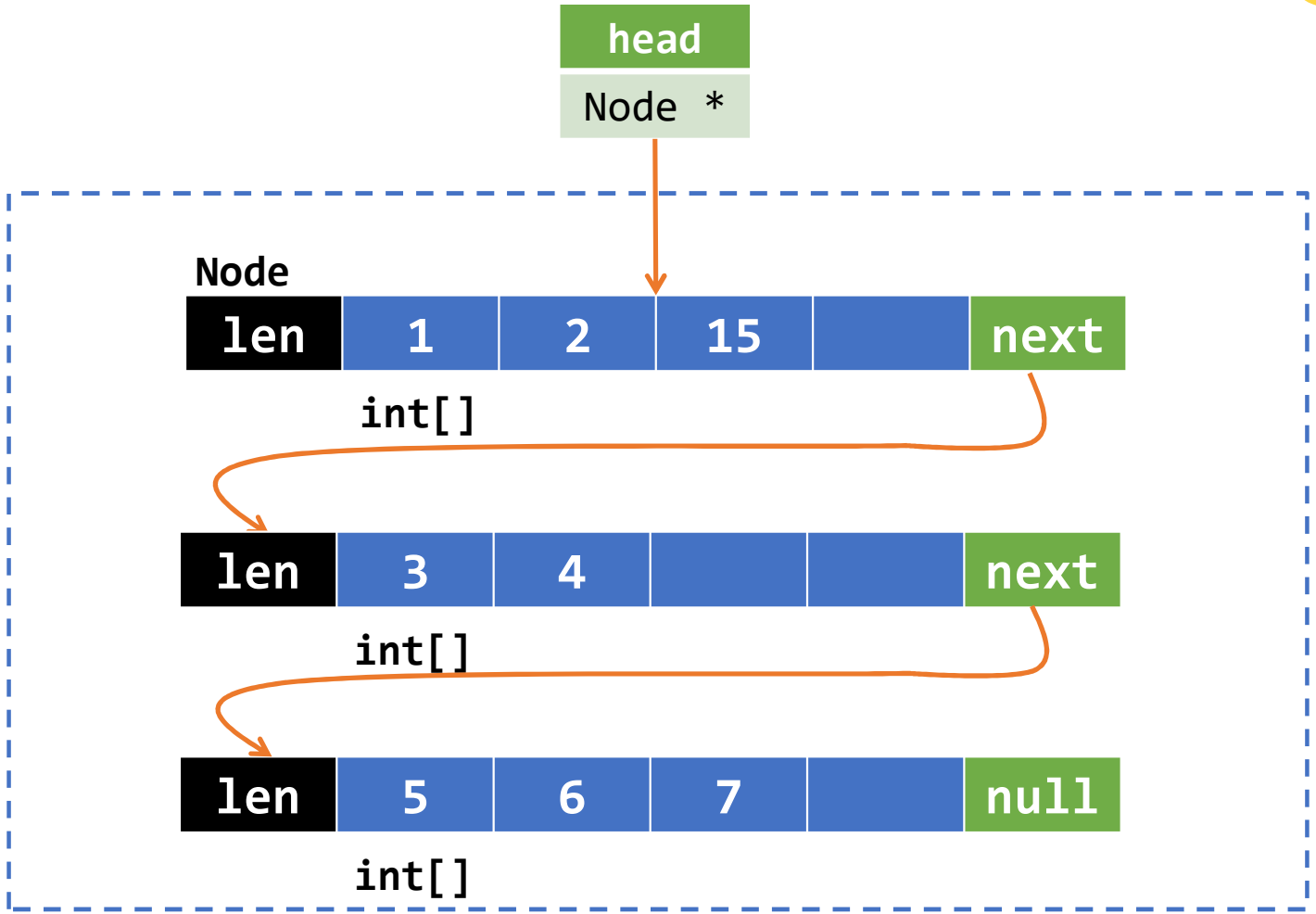


Развернутый список. Удаление значения



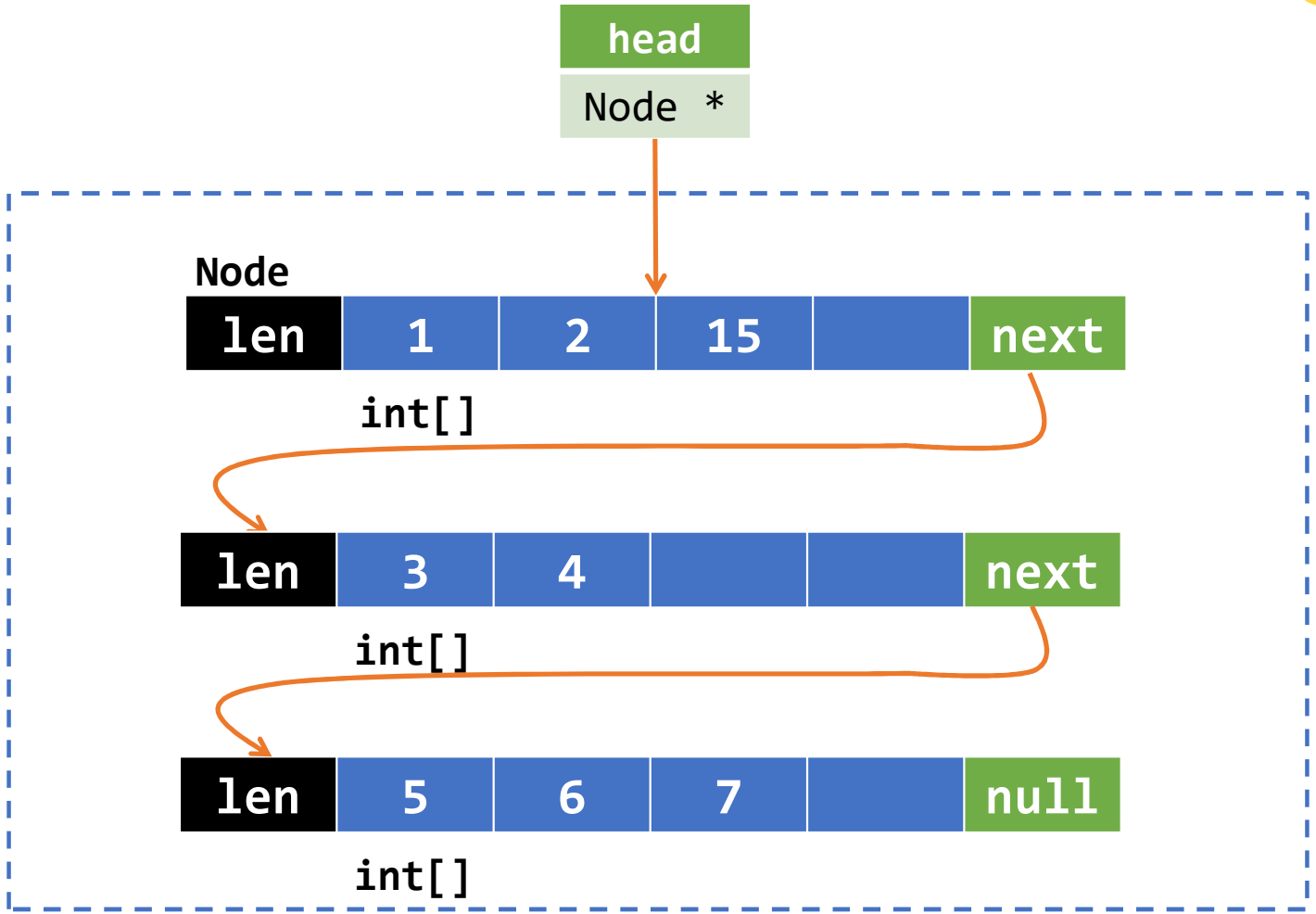
delete 8

Развернутый список. Удаление значения



delete 8

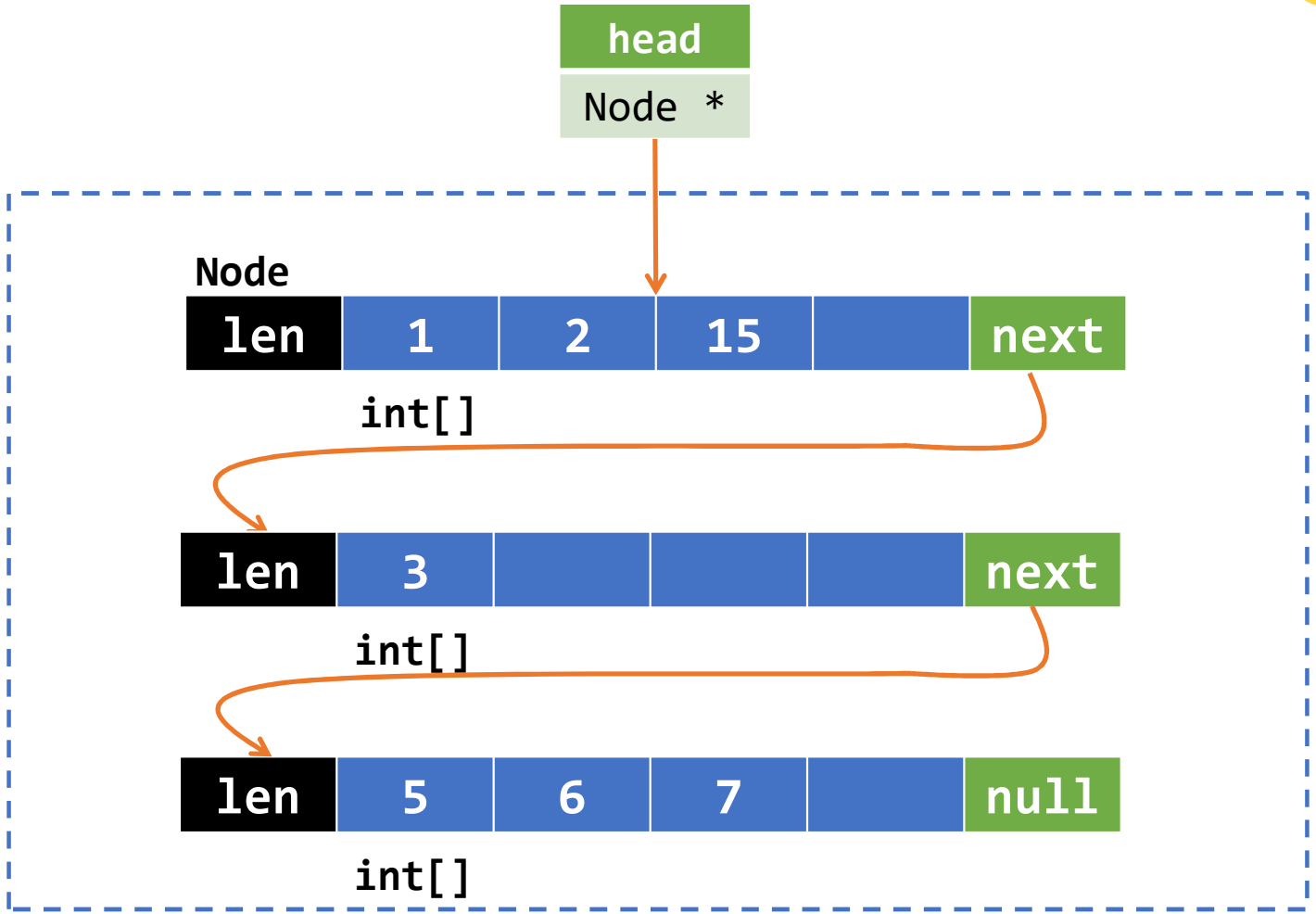
Развернутый список. Удаление значения



delete 8

delete 4

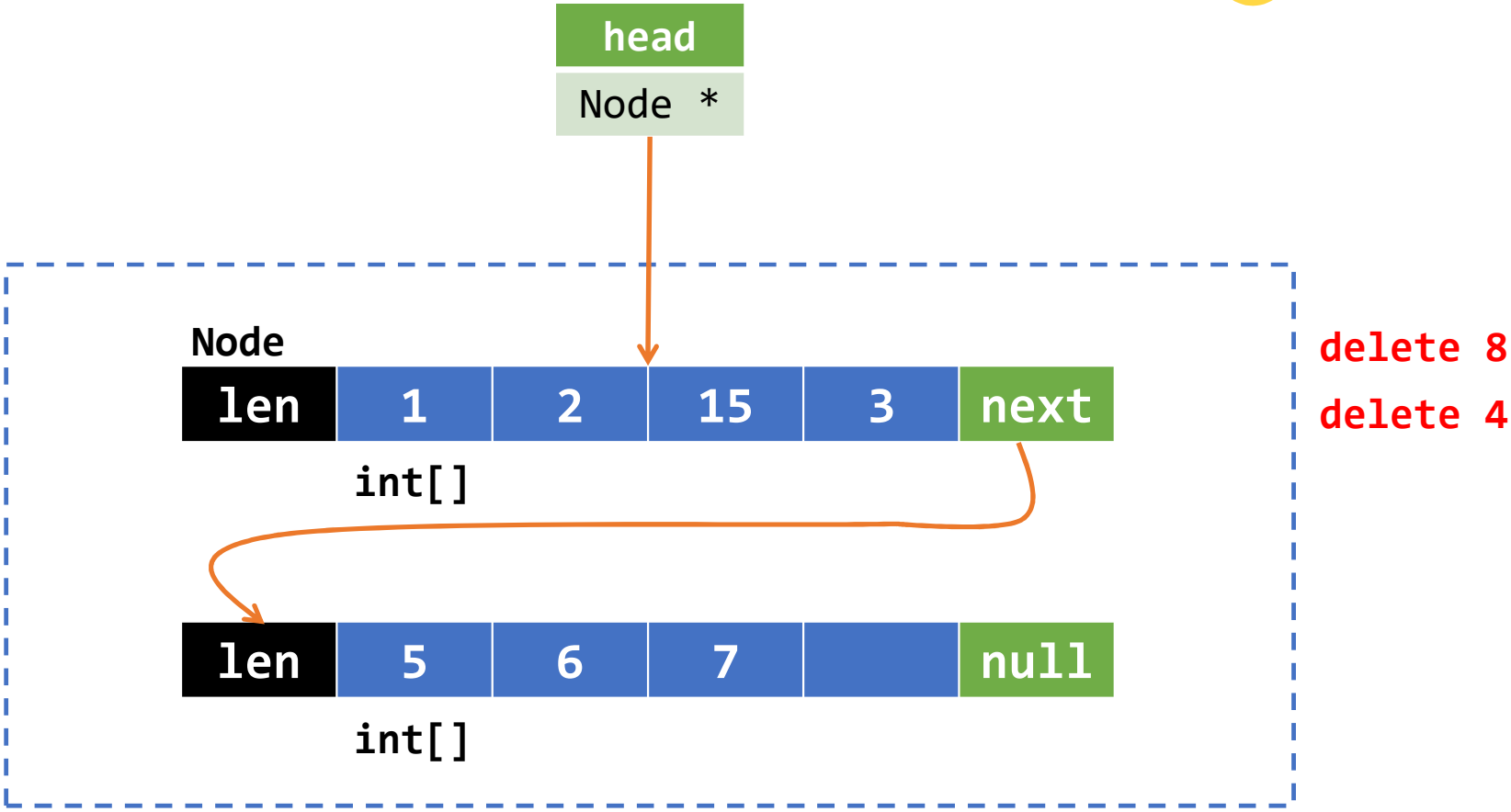
Развернутый список. Удаление значения



delete 8

delete 4

Развернутый список. Удаление значения



Развернутый список

- Более быстрое выполнение операций
- В отличие от обычных списков, относительный расход памяти для хранения указателей существенно ниже
- Основной вопрос при реализации – подбор размера массива, который хранится в узле списка