

Алгоритмы и Алгоритмические Языки

Семинар #14:

1. Структура данных “Очередь”.
2. Односвязный и двусвязный списки.
3. Кольцевой двусвязный список.
4. Кольцевой буфер.

Структура данных «Очередь»



Структура данных «Очередь»

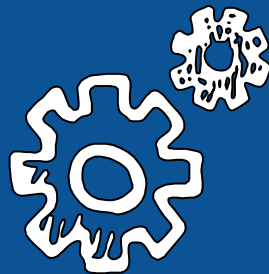
Задача – предоставление API, соответствующего реальной очереди.
Что можно делать с очередью:

- Добавление элемента в хвост очереди.
- Удаление элемента из головы очереди.
- Просмотр элемента из головы очереди (без удаления).

Возможный API:

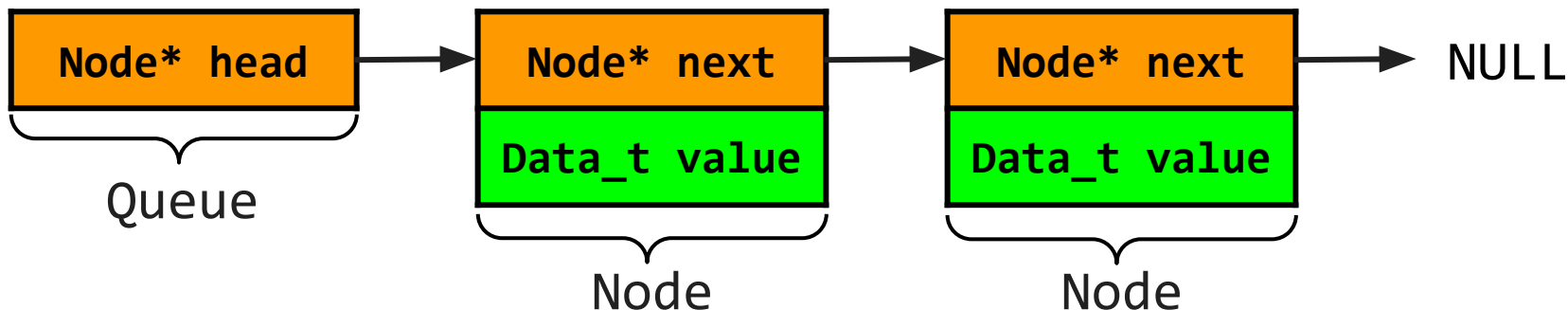
```
RetCode queue_alloc(Queue* queue);  
RetCode queue_free (Queue* queue);  
  
RetCode queue_peek      (Queue* queue, Data_t** data);  
RetCode queue_add_tail  (Queue* queue, const Data_t* data);  
RetCode queue_remove_head(Queue* queue, Data_t* data);  
bool    queue_empty     (Queue* queue);
```

Односвязный и двусвязный СПИСКИ



Односвязный список

Размещение односвязного списка в памяти:

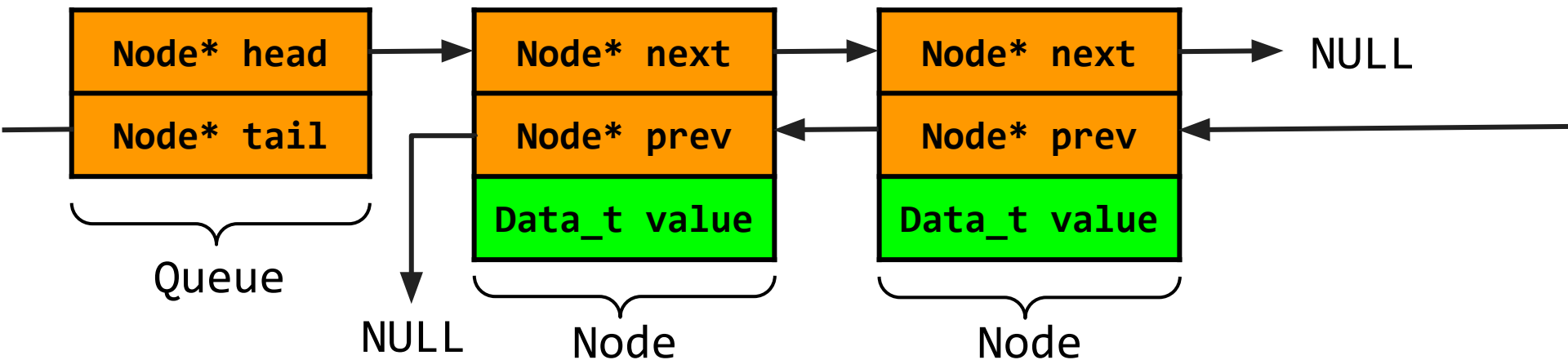


Вопросы:

1. Создание пустого односвязного списка.
2. Удаление односвязного списка.
3. Добавление элемента в хвост списка.
4. Удаление элемента из головы списка.

Двусвязный список

Размещение двусвязного списка в памяти:



Вопросы:

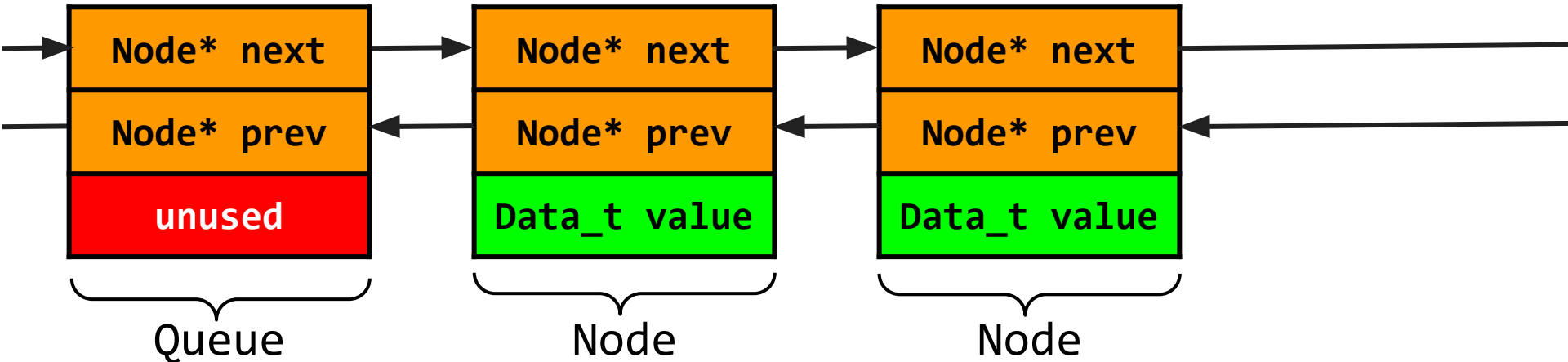
1. Добавление элемента в хвост списка.
2. Удаление элемента из головы списка.

Кольцевой двусвязный список



Кольцевой двусвязный список

Размещение кольцевого двусвязного списка в памяти:



Вопросы:

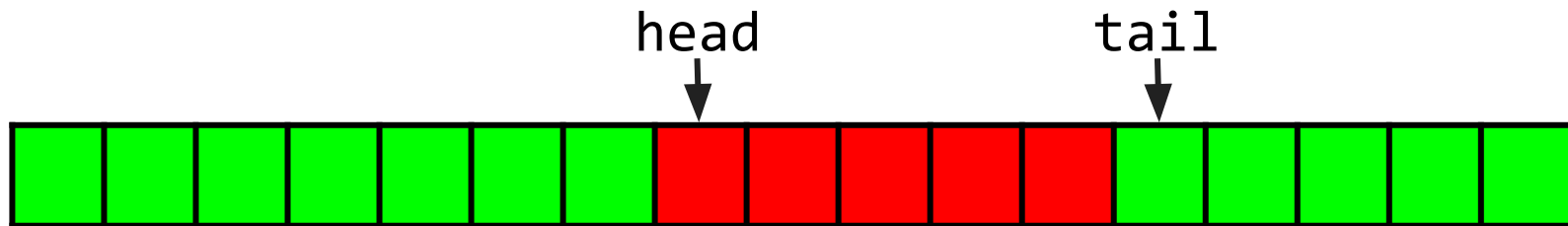
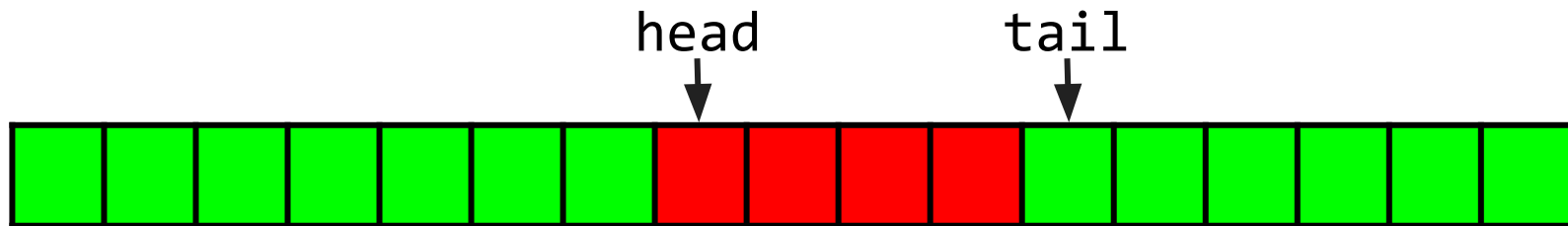
1. Добавление элемента в хвост списка.
2. Удаление элемента из головы списка.

Кольцевой буфер



Кольцевой буфер

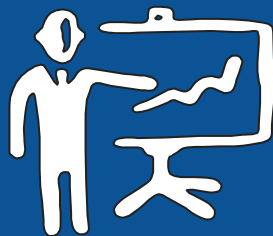
Реализация кольцевого буфера:



Тонкости:

- Надо знать максимальное кол-во элементов в очереди.
- Полная очередь vs пустая очередь.

Вопросы?



Красивые иконки взяты с сайта handdrawngoods.com