# K stack алгоритм с объединениями. Ссылки указывают только в одном направлении

Требование – хранить все данные в одном массиве. Ссылки на элементы указывают на более старые элементы. В вершине массива находятся более старые элементы. В конце болле новые элементы

Добавление элементы пачками [0;maxPerPage]

Elements = QList<Elements> //Массив элементов. Последний элемент массива новее по времени.

Top = QList<int> //Последние индексы в Массиве Elements каждого stack

Int currentStack

**Инициализация**

1. Запрос элементов
2. Вставка элементов в Elements
3. Вставка индекс верхнего элемента в Top (т.е. равно 0)
4. currentStack = 0

**Прокрутка** **вверх**

1. Запрос элементов
2. Вставка элементов сверху (в начало) по индексу currentStack
3. Обновление всех предыдущих Top[currentStack-1]
4. Проверка на соединения стеков. Если в начале текущего и конце предыдущего есть совпадения индексов элементов, то соединить.
5. Соединять не надо
   1. конец
6. Соединять надо
   1. Найти индекс h в текущем стеке элемент, совпадающий элементу предыдущего стека
   2. Удалить элементы [Top[currentStack]; h]
   3. S = Top[currentStack+1];Top.remove(currentStack+1);Top[currentStack]=S;

**Прокрутка вниз**

1. Запрос элементов
2. Вставка элементов снизу (в конец) по индексу currentStack. Номер Индекса равен Top[currentStack-1] или в конец Elements.
3. Обновление Top[currentStack-1] если currentStack>0
4. Проверка на соединения стеков. Если в конце текущего и начале предыдущего есть совпадения индексов элементов, то соединить.
5. Соединять не надо
   1. конец
6. Соединять надо
   1. Найти индекс h в текущем стеке элемент, совпадающий элементу предыдущего стека
   2. Удалить элементы [h;Top[currentStack-1]]
   3. S = Top[currentStack];Top.remove(currentStack);Top[currentStack-1]=S;
   4. currentStack--

**Переход по ссылке в несуществующий элемент (а если новая страница между стеками? Не проработал)**

1. Запрос элементов
2. Вставка элементов сверху (в начало) Elements
3. Вставка индекс верхнего элемента в Top
4. currentStack++
5. Обновление всех предыдущих Top[currentStack-1] (+ кол-во сообщений)

**Переход по ссылке в существующий элемент**

1. Поиск k подходящий номер стека по Top
2. Поиск индекса элемента в k
3. currentStack=k

|  |  |
| --- | --- |
| Message\_id | Ref\_id |
| 0 | 250 |
| 250 | 350 |
| 350 | 550 |
|  |  |
|  |  |
|  |  |

Тестовые сценарии:

1. [0:50] -> [0:600]
   1. Результат: currentStack=0;Top[currentStack]=0;Top.length==1;Elements.length==600
2. [550:600] -> [0:600]
   1. Результат: currentStack=0;Top[currentStack]=0;Top.length==1;Elements.length==600
3. [0:50],[250:300] ->[0:50],[200:300]
   1. Результат: currentStack=0;Top[currentStack]=0;Top[currentStack-1]=100;Top.length==2;Elements.length==150
4. [0:50],[250:300] ->[0:50],[200:300]->…-> [0:300]
   1. Результат: currentStack=0;Top[currentStack]=0;Top.length==1;Elements.length==300
5. [0:50],[250:300] ->[0:50],[250:600]
   1. Результат: currentStack=1;Top[currentStack]=0; Top[currentStack-1]=350;Top.length==2;Elements.length==400
6. [0:50],[250:300],[350:400],[550:600]-> [0:50],[250:400],[500:600]
   1. Результат: currentStack=1;Top[currentStack]=100; Top[currentStack-1]=250; Top[currentStack+1]=0;Top.length==3;Elements.length==300
7. [0:50],[250:300],[350:400],[550:600]-> [0:50],[250:300],[350:400],[500:600]->[0:50],[250:400](запросил и объединились),[500:600]-> [0:50],[200:400],[550:600]-> …->[0:400],[550:600]
   1. Результат: currentStack=0;Top[currentStack]=450;Top[currentStack+1]=0;Top.length==2;Elements.length==450
8. [0:50],[250:300],[350:400],[550:600]-> [0:50],[250:300],[350:400],[500:600]->[0:50],[250:400](запросил и объединились),[500:600]-> [0:50],[200:400],[550:600]-> …->[0:400],[550:600]-> …->[0:450],[550:600]->…-> [0:600]
   1. Результат: currentStack=0;Top[currentStack]=0;Top.length==1;Elements.length==600
9. [0:50],[250:300],[350:400],[550:600]-> [0:50],[250:400],[500:600]
   1. Результат: currentStack=2;Top[currentStack]=0; Top[currentStack-1]=100; Top[currentStack-2]=250;Top.length==3;Elements.length==300