

ОЧЕРЕДЬ –
ИЗ СТЕКОВ*Окончание. Начало в № 7 за 2023 г.*

– Итак, когда лифт всего один, получается несправедливо: чем раньше в него зашёл, тем позже выйдешь и позже попадёшь в столовую. Неужели можно исправить ситуацию, имея два лифта?

– Лёша, лифты в литературной студии подсказали мне идею! Там лифтов как раз два, «хороший» и «плохой». На «минус первый» этаж, где находится столовая, едет только хороший. Те, кто добираются в столовую на плохом лифте, выходят на первом этаже и переходят в хороший. Кто раньше зашёл в плохой лифт, позже войдёт в хороший и раньше окажется в очереди в столовую. То есть каждый лифт работает как стек, а два лифта – как очередь.

– Давай расскажу тебе историю о руководителе и его секретаре. В кабинете секретаря есть стопка прошений, куда каждый новый посетитель кладёт своё заявление сверху. На столе руководителя тоже есть стопка бумаг, которые он обрабатывает последовательно, начиная с верхнего. Когда вся стопка разобрана, руководитель зовёт секретаря, чтобы тот принёс следующие. Тогда секретарь берёт свою стопку и перекладывает её чиновнику на стол, но в обратном порядке, чтобы старые прошения оказались сверху. А теперь можешь рассказать итоговое решение?

– Думаю, да. Отличная история! Итак, есть два стека, и мы хотим добавлять и считывать элементы так, как в очереди. Назовём один стек входящим, а другой – исходящим. Новые элементы всегда добавляем во входящий стек. Считываем элементы всегда из исходящего стека.

– А если он пуст?

– Тогда перекладываем все элементы из входящего в исходящий, то есть последовательно вытаскиваем элементы из входящего стека, добавляем их в исходящий, пока входящий стек не станет пуст.

– Почему получилась очередь?

– Когда мы храним данные в стеке, то они «разворачиваются», то есть мы добавляем и считываем



элементы в противоположном порядке, это правило LIFO. А когда стеков два, данные разворачиваются дважды, и мы получим правило очереди – элемент, который добавили раньше другого, будет удалён из структуры тоже раньше этого самого другого.

– Отлично, с этой задачей разобрались. Тогда вот тебе ещё несколько. В них кроме операций push и pop, которые добавляют и извлекают (считывают с удалением) элементы из стека, можно использовать операцию read, которая считывает тот же элемент, что и операция pop, но не удаляет его. Сам этот элемент стека называется *верхним*. Ещё в задачах встречается понятие *правильной скобочной последовательности*, как думаешь, что это?

– Это как в правильном математическом выражении, только из него нужно удалить все числа и знаки действий?

– Да, можно такую модель держать в голове. Если формально, такая последовательность строится шаг за шагом по двум правилам. Начать можно с пары скобок, это правильная последовательность. Потом любую правильную последовательность можно заключить в скобки, получится снова правильная. Ещё можно две правильных последовательности записать рядом, это тоже правильная последовательность. Иногда скобки могут быть разного вида, например, круглые, квадратные или фигурные, тогда накладывается дополнительное правило – соответствующая закрывающаяся скобка должна быть того же типа, что и открывающаяся. А задачи такие:

1. Придумать структуру данных, в которой два стека, операции добавления и извлечения по правилу LIFO и операция readMin, которая считывает (но не удаляет) минимальный (в текущий момент) элемент среди имеющихся. Мы можем сравнивать элементы.

2. Мы последовательно считываем круглые скобки. Их можно хранить в одном стеке. Нужно определить, расставлены ли эти скобки правильно.

3. Та же задача, что и предыдущая, но теперь последовательность состоит из трёх видов скобок:

{}, [] и ().



Художник Мария Усеинова