Алгоритмы и структуры данных

Задание к лабораторной работе №4. Стек, очередь, связанный список.

Лабораторная работа посвящена разбору элементарных структур данных, таких как стек, очередь и связанный список. Аналогично предыдущим лабораторным работам, есть два способа ее выполнения и защиты:

Читать внимательно: изменения!

1 **Базовый уровень.** Решается 4 задачи по вариантам. Варианты в табличке внизу, номер вашего варианта соответствует вашему номеру в списке группы. Посмотреть свой номер можно, например, в журнале успеваемости по дисциплине. В этом случае максимум за защиту можно получить 4 балла. В сумме с самой работой (0,5 балла) и отчетом (1 балл) получается 5,5 балла, что достаточно для зачета.

2 **Продвинутый уровень.** Решаются все задачи или минимум 6 задач, причем 4 из них - исходя из вашего варианта, еще одна - из набора с 7 по 12, и как минимум 2 пункта 13й задачи. В этом случае вы сможете получить максимальные 7,5 баллов.

Вариант	Номера задач	Вариант	Номера задач
1	1,3,6,7	16	2,4,5,10
2	2,3,6,8	17	1,3,6,11
3	1,4,6,9	18	2,3,6,12
4	2,4,6,10	19	1,4,6,7
5	1,3,5,11	20	2,4,6,8
6	2,3,5,12	21	1,3,5,9
7	1,4,5,7	22	2,3,5,10
8	2,4,5,8	23	1,4,5,11
9	1,3,6,9	24	2,4,5,12
10	2,3,6,10	25	1,3,6,12
11	1,4,6,11	26	2,3,6,11
12	2,4,6,12	27	1,4,6,10
13	1,3,5,7	28	2,4,6,9
14	2,3,5,8	29	1,3,5,8
15	1,4,5,9	30	2,3,6,7

*Если задача вашего варианта из набора задач {7,8,9,10,11,12} вас по каким-то причинам не устраивает, вы можете выбрать другую задачу **из этого же набора**, или 13-ую. Возможно, 13-ая вам поможет в решении вашей задачи.*

1 задача. Стек

Реализуйте работу стека. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо "+ N", либо "–". Команда "+ N"означает добавление в стек числа N, по модулю не превышающего 10^9. Команда "–"означает изъятие элемента из стека. Гарантируется, что не происходит извлечения из пустого стека. Гарантируется, что размер стека в процессе выполнения команд не превысит 10^6 элементов.

- **Формат входного файла (input.txt).** В первой строке входного файла содержится M ($1 \leq M \leq 10^6$) – число команд. Каждая последующая строка исходного файла содержит ровно одну команду.
- **Формат выходного файла (output.txt).** Выведите числа, которые удаляются из стека с помощью команды "–", по одному в каждой строке. Числа нужно выводить в том порядке, в котором они были извлечены из стека. Гарантируется, что изъятий из пустого стека не производится.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Пример:

input.txt	output.txt
6 + 1 + 10 - + 2 + 1234 -	10 1234

2 задача. Очередь

Реализуйте работу очереди. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо «+ N», либо «–». Команда «+ N» означает добавление в очередь числа N, по модулю не превышающего 10^9. Команда «–» означает изъятие элемента из очереди. Гарантируется, что размер очереди в процессе выполнения команд не превысит 10^6 элементов.

- **Формат входного файла (input.txt).** В первой строке содержится M ($1 \leq M \leq 10^6$) – число команд. В последующих строках содержатся команды, по одной в каждой строке.
- **Формат выходного файла (output.txt).** Выведите числа, которые удаляются из очереди с помощью команды «–», по одному в каждой строке. Числа нужно выводить в том порядке, в котором они были извлечены из очереди. Гарантируется, что извлечения из пустой очереди не производится.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Пример:

input.txt	output.txt
4 + 1 + 10 - -	1 10

3 задача. Скобочная последовательность. Версия 1

Последовательность A, состоящую из символов из множества «(», «)», «[» и «]», назовем ***правильной скобочной последовательностью***, если выполняется одно из следующих утверждений:

- A – пустая последовательность;
- первый символ последовательности A – это «(», и в этой последовательности существует такой символ «)», что последовательность можно представить как $A = (B)C$, где B и C – правильные скобочные последовательности;
- первый символ последовательности A – это «[», и в этой последовательности существует такой символ «]», что последовательность можно представить как $A = (B)C$, где B и C – правильные скобочные последовательности.

Так, например, последовательности «(())» и «()[]» являются правильными скобочными последовательностями, а последовательности «[)» и «((» таковыми не являются.

Входной файл содержит несколько строк, каждая из которых содержит последовательность символов «(», «)», «[» и «]». Для каждой из этих строк выясните, является ли она правильной скобочной последовательностью.

- **Формат входного файла (input.txt).** Первая строка входного файла содержит число N ($1 \leq N \leq 500$) – число скобочных последовательностей, которые необходимо проверить. Каждая из следующих N строк содержит скобочную последовательность длиной от 1 до 10^4 включительно. В каждой из последовательностей присутствуют только скобки указанных выше видов.

- **Формат выходного файла (output.txt).** Для каждой строки входного файла (кроме первой, в которой записано число таких строк) выведите в выходной файл «YES», если соответствующая последовательность является правильной скобочной последовательностью, или «NO», если не является.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Пример:

input.txt	output.txt
5	YES
()()	YES
([])	NO
([)]	NO
(([]]	NO
)(	

4 задача. Скобочная последовательность. Версия 2

Определение правильной скобочной последовательности такое же, как и в задаче 3, но теперь у нас больше набор скобок: []{}().

Нужно написать функцию для проверки наличия ошибок при использовании разных типов скобок в текстовом редакторе типа LaTeX.

Для удобства, текстовый редактор должен не только информировать о наличии ошибки в использовании скобок, но также указать точное место в коде (тексте) с ошибочной скобочкой.

В первую очередь объявляется ошибка при наличии первой несовпадающей закрывающей скобки, перед которой отсутствует открывающая скобка, или которая не соответствует открывающей, например, ()[} - здесь ошибка укажет на }.

Во вторую очередь, если описанной выше ошибки не было найдено, нужно указать на первую несовпадающую открывающую скобку, у которой отсутствует закрывающая, например, (в ([].

Если не найдено ни одной из указанный выше ошибок, нужно сообщить, что использование скобок корректно.

Помимо скобок, код может содержать большие и маленькие латинские буквы, цифры и знаки препинания.

Формально, все скобки в коде (тексте) должны быть разделены на пары совпадающих скобок, так что в каждой паре открывающая скобка идет перед закрывающей скобкой, а для любых двух пар скобок одна из них вложена внутри другой, как в `(foo[bar])` или они разделены, как в `f(a,b)-g[c]`. Скобка [соответствует скобке], соответствует и (соответствует) .

- **Формат входного файла (input.txt).** Входные данные содержат одну строку S, состоящую из больших и маленьких латинских букв, цифр, знаков препинания и скобок из набора [] (). Длина строки $S - 1 \leq S\ leq 10^5$.

- **Формат выходного файла (output.txt).** Если в строке S скобки используются правильно, выведите «Success» (без кавычек). В противном случае выведите отсчитываемый от 1 индекс первой несовпадающей закрывающей скобки, а если нет несовпадающих закрывающих скобок, выведите отсчитываемый от 1 индекс первой открывающей скобки, не имеющей закрывающей.
- Ограничение по времени. 5 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt
[]	Success
{}[]	Success
[()]	Success
(())	Success
{	1
{[}	3
foo(bar);	Success
foo(bar[i);	10

5 задача. Стек с максимумом

Стек - это абстрактный тип данных, поддерживающий операции `Push()` и `Pop()`. Нетрудно реализовать его таким образом, чтобы обе эти операции работали за константное время. В этой задаче ваша цель - реализовать стек, который также поддерживает поиск максимального значения и гарантирует, что все операции по-прежнему работают за константное время.

Реализуйте стек, поддерживающий операции `Push()`, `Pop()` и `Max()`.

- **Формат входного файла (input.txt).** В первой строке входного файла содержится n ($1 \le n \le 400000$) – число команд. Последующие n строк исходного файла содержит ровно одну команду: `push V`, `pop` или `max`. $0 \le V \le 10^5$.
- **Формат выходного файла (output.txt).** Для каждого запроса `max` выведите (в отдельной строке) максимальное значение стека.
- Ограничение по времени. 5 сек.
- Ограничение по памяти. 512 мб.
- Пример:

input.txt	output.txt	input.txt	output.txt	input.txt	output.txt
5 push 2 push 1 max pop max	2 2	5 push 1 push 2 max pop max	2 1	3 push 1 push 7 pop	

input.txt	output.txt	input.txt	output.txt
10	9	6	7
push 2	9	push 7	7
push 3	9	push 1	
push 9	9	push 7	
push 7		max	
push 2		pop	
max		max	
max			
max			
pop			
max			

6 задача. Очередь с минимумом

Реализуйте работу очереди. В дополнение к стандартным операциям очереди, необходимо также отвечать на запрос о минимальном элементе из тех, которые сейчас находится в очереди. Для каждой операции запроса минимального элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда – это либо «+ N», либо «–», либо «?». Команда «+ N» означает добавление в очередь числа N, по модулю не превышающего 10^9. Команда «–» означает изъятие элемента из очереди. Команда «?» означает запрос на поиск минимального элемента в очереди.

- **Формат входного файла (input.txt).** В первой строке содержится M ($1 \leq M \leq 10^6$) – число команд. В последующих строках содержатся команды, по одной в каждой строке.
- **Формат выходного файла (output.txt).** Для каждой операции поиска минимума в очереди выведите её результат. Результаты должны быть выведены в том порядке, в котором эти операции встречаются во входном файле. Гарантируется, что операций извлечения или поиска минимума для пустой очереди не производится.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Пример:

input.txt	output.txt
7 + 1 ? + 10 ? - ? -	1 1 10

- Вам может помочь идея, изложенная во второй части вот этой страницы.

7 задача. Максимум в движущейся последовательности

Задан массив из n целых чисел - $a_1, ..., a_n$ и число $m < n$, нужно найти максимум среди последовательности ("окна") $\{a_i, ..., a_{i+m-1}\}$ для каждого значения $1 \leq i \leq n - m + 1$. Простой алгоритм решения этой задачи за $O(nm)$ сканирует каждое "окно"отдельно.

Ваша цель - алгоритм за $O(n)$.

- **Формат входного файла (input.txt).** В первой строке содержится целое число n ($1 \leq n \leq 10^5$) – количество чисел в исходном массиве, вторая строка содержит n целых чисел $a_1, ..., a_n$ этого массива, разделенных пробелом ($0 \leq a_i \leq 10^5$). В третьей строке - целое число m - ширина "окна"($1 \leq m \leq n$).
- **Формат выходного файла (output.txt).** Нужно вывести $\max a_i, ..., a_{i+m-1}$ для каждого $1 \leq i \leq n - m + 1$.
- Ограничение по времени. 5 сек.
- Ограничение по памяти. 512 мб.
- Пример:

input.txt	output.txt
8 2 7 3 1 5 2 6 2 4	7 7 5 6 6

- Есть несколько решений этой задачи. Например:
  - использование очереди на основе двух стеков;
  - использование Dequeue.

8 задача. Постфиксная запись

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как A B +. Запись B C + D * обозначает привычное нам (B + C) * D, а запись A B C + D * + означает A + (B + C) * D. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Дано выражение в обратной польской записи. Определите его значение.

- **Формат входного файла (input.txt).** В первой строке входного файла дано число N ($1 \leq n \leq 10^6$) – число элементов выражения. Во второй строке содержится выражение в постфиксной записи, состоящее из N элементов. В выражении могут содержаться неотрицательные однозначные числа и операции +, -, *. Каждые два соседних элемента выражения разделены ровно одним пробелом.
- **Формат выходного файла (output.txt).** Необходимо вывести значение записанного выражения. Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений, по модулю будут меньше, чем 2^{31}.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Пример:

input.txt	output.txt
7 8 9 + 1 7 - *	-102

9 задача. Поликлиника

Очередь в поликлинике работает по сложным правилам. Обычные пациенты при посещении должны вставать в конец очереди. Пациенты, которым "только справку забрать встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром. Напишите программу, которая отслеживает порядок пациентов в очереди.

- **Формат входного файла (input.txt).** В первой строке записано одно целое число n ($1 \leq n \leq 10^5$) - число запросов к вашей программе. В следующих n строках заданы описания запросов в следующем формате:
  - «+ i» – к очереди присоединяется пациент i ($1 \leq i \leq N$) и встает в ее конец;
  - «* i» – пациент i встает в середину очереди ($1 \leq i \leq N$);
  - «-» – первый пациент в очереди заходит к врачу. Гарантируется, что на момент каждого такого запроса очередь будет не пуста.

- **Формат выходного файла (output.txt).** Для каждого запроса третьего типа в отдельной строке выведите номер пациента, который должен зайти к шаманам.

- Ограничение по времени. Оцените время работы и используемую память при заданных максимальных значениях.

- Пример:

input.txt	output.txt	input.txt	output.txt
7 + 1 + 2 - + 3 + 4 - -	1 2 3	10 + 1 + 2 * 3 - + 4 * 5 - - - -	1 3 2 5 4

10 задача. Очередь в пекарню

В пекарне работает один продавец. Он тратит на одного покупателя ровно 10 минут, а затем сразу переходит к следующему, если в очереди кто-то есть, либо ожидает, когда придет следующий покупатель. Даны времена прихода покупателей в пекарню (в том порядке, в котором они приходили). Также у каждого покупателя есть характеристика, называемая *степенью нетерпения*. Она показывает, сколько человек может максимально находиться в очереди перед покупателем, чтобы он дождался своей очереди и не ушел раньше. Если в момент прихода покупателя в очереди находится больше людей, чем степень его нетерпения, то он решает не ждать своей очереди и уходит. Покупатель, который обслуживается в данный момент, также считается находящимся в очереди. Требуется для каждого покупателя указать время его выхода из пекарни.

- **Формат входного файла (input.txt).** В первой строке вводится натуральное число N, не превышающее 100 - количество покупателей. В следующих N строках вводятся времена прихода покупателей - по два числа, обозначающие часы и минуты (часы - от 0 до 23, минуты - от 0 до 59) и степень его нетерпения (неотрицательное целое число не большее 100) - максимальное количество человек, которое он готов ждать впереди себя в очереди. Времена указаны в порядке возрастания (все времена различны). Гарантируется, что всех покупателей успеют обслужить до полуночи. Если для каких-то покупателей время окончания обслуживания одного покупателя и время прихода другого совпадают, то можно считать, что в начале заканчивается обслуживание первого покупателя, а потом приходит второй покупатель.

- **Формат выходного файла (output.txt).** В выходной файл выведите N пар чисел: времена выхода из пекарни 1-го, 2-го, ..., N-го покупателя (часы и минуты). Если на момент прихода покупателя человек в очереди больше, чем степень его нетерпения, то можно считать, что время его ухода равно времени прихода.

- Ограничение по времени. Оцените время работы и используемую память при заданных максимальных значениях.

- Пример:

input1.txt	output1.txt	input2.txt	output2.txt
3	10 10	5	13 10
10 0 0	10 20	13 0 100	14 10
10 1 1	10 2	14 0 0	14 1
10 2 1		14 1 0	14 20
		14 2 3	14 3
		14 3 0	

11 задача. Бюрократия

В министерстве бюрократии одно окно для приема граждан. Утром в очередь встают n человек, i-й посетитель хочет получить a_i справок. За один прием можно получить только одну справку, поэтому если после приема посетителю нужны еще справки, он встает в конец очереди. За время приема министерство успевает выдать m справок. Остальным придется ждать следующего приемного дня. Ваша задача - сказать, сколько еще справок хочет получить каждый из оставшихся в очереди посетитель в тот момент, когда прием закончится. Если все к этому моменту разойдутся, выведите -1.

- **Формат входного файла (input.txt).** В первой строке - количество посетителей n ($1 \leq n \leq 10^5$) и количество справок m ($0 \leq m \leq 10^9$). Во второй строке для каждого посетителя в порядке очереди указано количество справок a_i ($1 \leq ai \leq 10^6$), которое он рассчитывает получить. Номером посетителя называется его место в исходной очереди.

- **Формат выходного файла (output.txt).** В первой строке выведите, сколько посетителей останется в очереди, когда прием закончится. Во второй строке выведите состояние очереди на тот момент, когда прием закончится: для всех посетителей по порядку выведите по одному числу через пробел - количество справок, которое он хочет еще получить. В случае, если в очереди никого не останется выведите одно число: -1

- Ограничение по времени. Оцените время работы и используемую память при заданных максимальных значениях.

- Пример:

input1.txt	output1.txt	input2.txt	output2.txt
3 2 1 2 3	2 3 1	4 5 2 5 2 3	3 4 1 2

12 задача. Строй новобранцев

В этой задаче n новобранцев, пронумерованных от 1 до n, разделены на два множества: *строй* и *толпа*. Вначале *строй* состоит из новобранца номер 1, все остальные составляют *толпу*. В любой момент времени строй стоит в один ряд по прямой. Товарищ сержант может использовать четыре команды. Вот они.

- "I, встать в строй слева от J."Эта команда заставляет новобранца номер I, находящегося в толпе, встать слева от новобранца номер J, находящегося встрою.
- "I, встать в строй справа от J."Эта команда действует аналогично предыдущей, за исключением того, что I встает справа от J.
- "I, выйти из строя."Эта команда заставляет выйти из строя новобранца номер I. После этого он присоединяется ктолпе.
- "I, назвать соседей."Эта команда заставляет глубоко задуматься новобранца номер I, стоящего в строю, и назвать номера своих соседей по строю, сначала левого, потом правого. Если кто-то из них отсутствует (новобранец находится на краю ряда), то вместо соответствующего номера он должен назвать 0.

Известно, что ни в каком случае строй не остается пустым. Иногда строй становится слишком большим, и товарищ сержант уже не может проверять сам, правильно ли отвечает новобранец. Поэтому он попросил вас написать программу, которая помогает ему в нелегком деле обучения молодежи и выдает правильные ответы для его команд.

- **Формат входного файла (input.txt).** В первой строке находятся два числа N ($1 \leq N \leq 75000$) и M ($1 \leq M \leq 75000$) - количество новобранцев и команд соответственно. Следующие M строк содержат команды, одна команда на строку. Каждая команда - одна из следующих:
  - `left I J` - соответствует команде "I, встать в строй слева от J."
  - `right I J` - "I, встать в строй справа от J."
  - `leave I` - "I, выйти из строя."
  - `name I` - "I, назвать соседей."

  Гарантируется, что все команды корректны, например, `leave I` не будет заставлять выйти из строя новобранца, стоящего в толпе. Также гарантируется, что строй никогда не будет пустым.

- **Формат выходного файла (output.txt).** Для каждой строки, содержащей `name I`, выведите в отдельной строке два числа - номера левого и правого соседа новобранца номер I. Если кто-то из соседей отсутствует, выведите ноль вместо его номера.
- Ограничение по времени. Оцените время работы и используемую память при заданных максимальных значениях.
- Примеры:

input1.txt	output1.txt	input2.txt	output2.txt
3 3 left 2 1 right 3 1 name 1	2 3	3 4 left 2 1 right 3 1 leave 1 name 2	0 3

13 задача★. Реализация стека, очереди и связанных списков

1. Реализуйте стек на основе связного списка с функциями isEmpty, push, pop и вывода данных.
2. Реализуйте очередь на основе связного списка функциями Enqueue, Dequeue с проверкой на переполнение и опустошения очереди.
3. Реализуйте односвязный список с функциями вывода содердимого списка, добавления элемента в начало списка, удаления элемента с начала списка, добавления и удаления элемента *после* заданного элемента (key); поиска элемента в списке.
4. Реализуйте двусвязный список с функциями вывода содердимого списка, добавления и удаления элемента с начала списка, добавления и удаления элемента с конца списка, добавления и удаления элемента *до* заданного элемента (key); поиска элемента в списке.