

Результаты подсчета занимаемой памяти переменными для урока №6 Алгоритмы
 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) - версия python
 MSC v.1927 64 bit (AMD64) win32 - платформа

Реализация	var1 использование строк			var2 использование списков			var3 использование очередей		
строка(справа) <hr/> структура данных(снизу)	'12'	'123 456 789 0'	50 символов	'12'	'1234567890'	50 символов	'12'	'1234567890'	50 символов
num type= <class 'int'>	24	24	24	24	24	28	24	24	28
num_copy type= <class 'int'>	28	32	48	28	32	48	28	32	48
digit type= <class 'int'>	28	28	28	28	28	28	28	28	28
even	str 50	str 54	str 70	list 88	list 120	list 248	deque 624	deque 1152	deque 1152
сумма элементов even	-	-	-	28	136	560	28	136	560
uneven	str 50	str 54	str 78	list 88	list 120	list 312	deque 624	deque 1152	deque 1152
сумма элементов uneven	-	-	-	28	140	840	28	140	840
сумма	180	192	248	312	600	2064	1384	2664	3808

(все цифры представлены в байтах)

Выводы: наиболее оптимальным решением использовать строковые данные для данной задачи, так как строки по сравнению со списком и очередью требовали значительно меньше компьютерных ресурсов. Кроме того, строки, с увеличением количества данных в 5 раз (для 10 символов), памяти потребовалось всего на 6% (180 и 192) больше. Тогда как увеличение данных в 5 раз (для 10 символов) для списков потребовало на 100% (312 и 600) больше ресурсов, как и для очередей. Дальнейшее увеличение количества данных в 5 раз (50 символов), память для строк увеличилась на 25% (248 и 192). Списки - память увеличилась в 3,3 раза (600 и 2064), очереди - на 40% (2664 и 3808).

Сложно по этим данным проследить какую-то тенденцию, надо делать больше замеров для разного количества данных. Однако ясно видно, с использованием строк для решения задачи, память используется наиболее эффективно.