**1. Question 1**

Download the following text file:

[Median.txt](https://d3c33hcgiwev3.cloudfront.net/_6ec67df2804ff4b58ab21c12edcb21f8_Median.txt?Expires=1523491200&Signature=HY35Z8m7OwIYCcK0iKo9VDUqQe~B3uLEWqngjynBgjqDrkjtJbtMhNi2pmtFM0t9t1cXjXwZCj7iE4C4CQCFK18DHj6pGEPVsjOiMw3f3QhiMk6~1PSZizQm2soDU6zWix6yGaGEt9dXC5aKoXqsWReaQh7tSV2umwh3jgZX~5w_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)



The goal of this problem is to implement the "Median Maintenance" algorithm (covered in the Week 3 lecture on heap applications). The text file contains a list of the integers from 1 to 10000 in unsorted order; you should treat this as a stream of numbers, arriving one by one. Letting xi denote the ith number of the file, the kth median mk is defined as the median of the numbers x1,…,xk. (So, if k is odd, then mk is ((k+1)/2)th smallest number among x1,…,xk; if k is even, then mk is the (k/2)th smallest number among x1,…,xk.)

In the box below you should type the sum of these 10000 medians, modulo 10000 (i.e., only the last 4 digits). That is, you should compute (m1+m2+m3+⋯+m10000)mod10000.

OPTIONAL EXERCISE: Compare the performance achieved by heap-based and search-tree-based implementations of the algorithm.