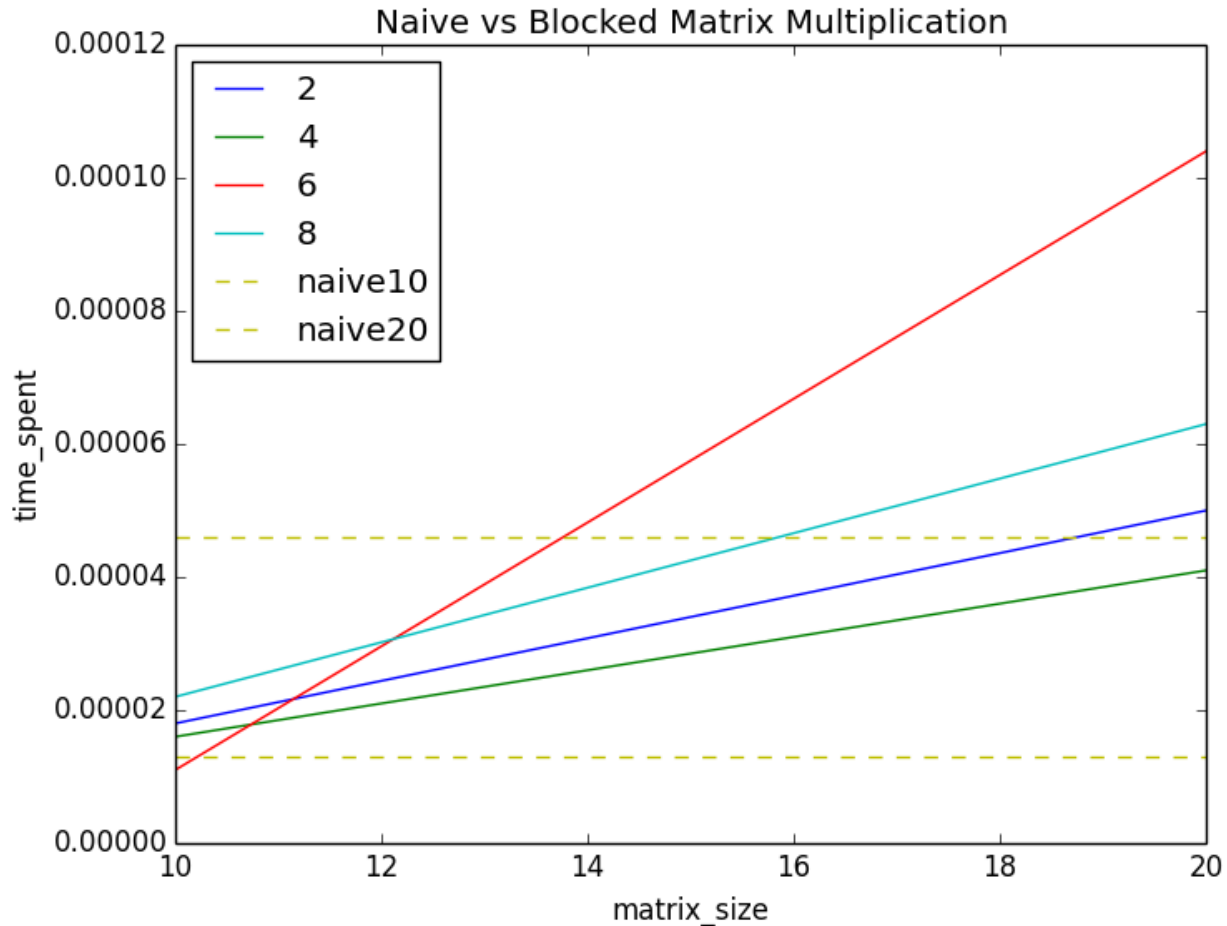


Assignment1

Question 1

For the matrix with smaller sizes naive_matrix_multiplication method is faster, but as size of matrix increases we observe that blocked_matrix_multiplication is faster.



Question 2

Bitmap representations allow fastest algorithm as compared to index representations. In frequent pairs case, bitmap representation provide solution in **average 153 secs** , while index representation takes **average 462 secs** over bow.dtm dataset.

(While for threshold 200, the algorithm can be as fast as 30 sec in case of index representation, as I drop all the vocab words for which length of index_list is less than threshold as they can not be in output, its just a trick I used to not process unnecessary vocab words which cant be in output set eventually, so no need to process them. But for worst case, threshold 1, it takes avg 462 secs)

Hence, bitmap representation allows faster algorithm.

Algorithms can be optimized further from their naive cases by using better algorithms and search techniques. And later on, SIMD can be used to push the efficiency of a single core.

Which representation is better depends on problem in hand. Hence, there can be circumstances where index representation can be much efficient compared to bitmap representation. Suppose, we want to know the frequency of a vocab word (number of documents a word appears in), in this case, index representation would provide results in constant time ($O(1)$) for each vocab word, but for bitmap representation, we will have to find the sum of all set bits for each vocab word, which will not be constant time lookup.

Other data representations are possible, but they may not be as efficient as bitmap or index representations. Hence, better data representations are not possible in my opinion for the given case.