X1

a)

We will proceed using a method very similar to multi-pass hash joining. Let V represent the Visit table. Partition V using a hash function on uid. The first pass will only be able to divide V into $M-1$ partitions. Subsequent passes will divide these partitions into $M-1$ subpartitions. This algorithm will perform enough partitions until each partition is able to fit inside memory that is, when each partitions takes up $M-1$ blocks or less. At this point, the algorithm reads in each partition, and iterates through the rows tallying the frequency that each uid occurs. Because we hashed based on uid, all occurrences of a given uid will be in the same partition, though there might be multiple uids in that partition. We will also keep track of the highest number of visits and a list of the uids that have this frequency of visits. After reading and counting the partition, we will update the max accordingly and discard all uids below the new maximum frequency.

The number of I/O's depends on the number of passes used to recursively partition V. Let $B(V)$ represent the number of blocks for all of V. At each pass we read in the total number of blocks and write it back to memory again for $2B(V)$ total I/O's, and there is an additional read-in at the end to count frequencies. V has 25.6 million rows and there are 8 bytes per row, for 204.8 million bytes total. Since there are 8192 bytes per block, this becomes 25,000 blocks. The number of passes needed to create partitions of size at most $M-1=9$ blocks is $\left\lceil \log_{M-1}\lceil\frac{B(V)}{M}\rceil\right\rceil + 1 = \lceil \log_9\lceil\frac{25000}{10}\rceil\rceil + 1 = 5$. So the total number of I/O's needed is $2B(V)\cdot 5 + B(V) = 2(25000)(5) + 25000 = 275000$ I/O's.

b)

Use the same algorithm as in a) except in each pass only partition into $M-2$ partitions. The additional block will be used to keep track of the number of rows in each partition. After each pass, if any of the partitions contain fewer rows than the number of visits we are interested in, which in this case is 3000, we will no longer consider that partition in the future.