Algorithm 3: Segmentation and sorting of data blocks

**Input:** splits, prioritySet  
**Output:** prioritySplits

1. prioritySplits ← an empty list
2. for i = 0; i < splits.size; i++ do
3.     if prioritySet.contains(i) then
4.         prioritySplits.addAll(getSmall(splits.get(i)), HIGH_LEVEL)
5.     else
6.         prioritySplits.addAll(splits.get(i), NORMAL_LEVEL)
7.     end if
8. end for
9. prioritySplits are sorted by priority from high to low
10. return prioritySplits

The algorithm description for segmenting data blocks and sorting is shown as algorithm 3. The original data blocks are traversed sequentially and if the index number of the block exists in the prioritySet, it is finely divided into N new blocks which are set to high priority. The rest that are not in the prioritySet are not segmented and set to standard priority. After traversing all the data blocks, they will be sorted by their priority, thereby ensuring that time-consuming blocks will be processed firstly.