Hints for OnlineJudge 11413:

This problem is phrased in an awkward way, but is asking something relatively simple. The input has a sequence of numbers (representing the amount of milk in each vessel). You need to partition these numbers into contiguous blocks, e.g., 1,2,3,4,5,6,1,7,2 could be partitioned into {1,2,3,4} {5,6} {1,7,2} (each block represents combining those vessels of milk into a single container). You want the "biggest" block (the one with largest sum) to be as small as possible.

This can be solved by binary search, not over the input data but over the *optimal value* you're looking for. Suppose we have a guess, call it "mid", for the optimum. To tell whether the true optimum is <= mid or > mid, we just need to know whether it's possible to partition the numbers into m contiguous blocks, each having sum <= mid. This can be done greedily: just think of the leftmost numbers as going to the first block until their sum exceeds mid, then start a new block and add as many numbers (continuing left-to-right) as possible until that sum exceeds mid, and so on. If processing all n numbers this way creates at most m blocks, then we should restrict our binary search to have mid as the new upper bound. Otherwise, it's not possible, so we should let mid+1 be the new lower bound.

The initial possible range for the binary search is from (the maximum input number) to (the sum of all input numbers).