Split (Review)

Subtask 1

In this subtask, the given graph is a path or a cycle. Therefore, the answer is always YES and a solution can be easily found by partitioning a path: $v_1, v_2, \ldots, v_n$ into $A = [v_1, v_2, \ldots, v_a]$, $B = [v_{a+1}, v_{a+2}, \ldots, v_{a+b}]$, and $C = [v_{a+b+1}, v_{a+b+2}, \ldots, v_n]$. In case of a cycle, we can cut the cycle in any place and apply the similar solution as we had in path.

Subtask 2

In this subtask, the answer is always YES as follows. Let us assume that $a \leq b \leq c$. Find a connected subgraph of size b (like using DFS or any graph traversing algorithm) as subset B, construct subset A consisting of an arbitrary vertex outside of B, and other vertices are set C.

Subtask 3

In this subtask, the given graph is a tree. The solution is similar to that of the the last subtask but the cases to be considered are easier. Without loss of generality suppose $a \leq b \leq c$. One solution is to run a DFS on an arbitrary vertex to find some arbitrary spanning tree and find a vertex v such that the size of the subtree of v [including v] denoted by $size(v)$ is at least a, but the sizes of the subtrees of all children of v are less than a. Remove the edge between v and the parent of v which partition the tree into two connected components. If both of the components have a size more than a, then the answer is YES. Otherwise, the answer is no since removing v from the original graph only produces components of size less than a. If the answer is YES, then a connected subgraph of size a (namely A) from the smallest component, and a connected subgraph of size b (namely B) from the smallest component can be extracted (since $b \leq c$, the size of the larger component is at least b). Finally, C consists of all of the remaining vertices.

Subtask 4

The input of this subtask is similar to that of the final subtask. However, since the limits for n and m are more restricted, one can use n instances of DFS instead of only one.

Subtast 5

Similar to the solution of Subtask 3, assume we run a DFS on an arbitrary vertex and found a vertex v such that $size(v) \geq a$ but the sizes of the subtrees of the children of v are all smaller than a. Note that other edges outside the DFS tree are backward edges. Hence, the children of v may have backward edges to v or the ancestors of v, but no edge exists between them. Suppose we want to partition the graph into two connected components by only considering the edges of the DFS tree and removing the edge between v and the parent of v. In this way the graph is partitioned into part P_1 consisting of v and its subtree and P_2 consisting of all other vertices. Before proceeding, we check every child of v (in an arbitrary order) like u and if the following conditions hold, we remove its subtree from P_1 and add it to P_2. The subtree of u has a backward edge to the ancestors of v. The size of P_1 after removing the subtree of u is still at least a. By doing so, one by one, we may encounter a child of v, namely u, such that $|P_1| \geq a$ and $|P_1| - size(u) < a$. Hence, $|P_2| > (n - 2a) = (a + b + c - 2a) = (b + c - a) \geq b$. In the other case, either $|P_2| > a$, which is also good since $|P_1| > n - b = (a + b + c) - b = a + c > b$, or $|P_2| < a$ and the answer is NO, since v is a cut vertex such that removing it from the original graph produces several components where all of them are of size less than a.