1. Yes, because expressions like {1,2} | myset require your type to handle the case where the left operand is a built-in set and your type is on the right. __ror__ lets the right operand respond when the left operand does not know how to handle the operation.
2. Yes, because |= is meant to be an in-place update for sets and dicts, and supporting __ior__ lets your type perform the mutation efficiently. Without __ior__, |= would fall back to making a new object instead of updating the existing one.