

```
/**
 * hasValidNetwork() determines whether "this" GameBoard has a valid network
 * for player "side". (Does not check whether the opponent has a network.)
 * A full description of what constitutes a valid network appears in the
 * project "README" file.
 *
 * Unusual conditions:
 *   If side is neither MachinePlayer.COMPUTER nor MachinePlayer.OPPONENT,
 *   returns false.
 *   If GameBoard squares contain illegal values, the behavior of this
 *   method is undefined (i.e., don't expect any reasonable behavior).
 *
 * @param side is MachinePlayer.COMPUTER or MachinePlayer.OPPONENT
 * @return true if player "side" has a winning network in "this" GameBoard;
 *         false otherwise.
 */
protected boolean hasValidNetwork(int side)
```