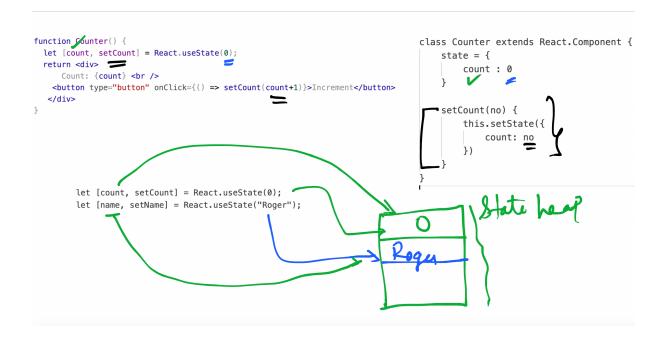


shouldComponentUpdate();



```
function counterReducer(state, action) {
       \textbf{switch}(\textbf{action.type}) \ \{
         case 'INCREMENT':
                                                                                                                        count: 0
           return {
              count: state.count + action.payload
         case 'DECREMENT':
           return {
              count: state.count - 1
         case 'RESET':
           return {
              count: 0
                                                             function Sample() {
                                                               let [state, dispatch] = React.useReducer(counterReducer, {count:0});
                                                               return <div>
Count: {state.count} <br />
                                                                 </div>
                                                                                           13
function counterReducer(state, action) {
  {\color{red} \textbf{switch}}(\textbf{action.type}) \ \{
                                                1A
     case 'INCREMENT':
        return {
                                                                                                                           count: 0
          count: state.count + action.payload
     case 'DECREMENT':
       return {
         count: state.count - 1
     case 'RESET':
       return {
         count: 0
                                                            function Sample() {
                                                              let [state, dispatch] = React.useReducer(counterReducer, {count:0});
}
                                                               return <div>
                                                                   Count: {state.count} <br />
                                                                 <button type="button" onClick={() => dispatch({type:'INCREMENT', payload: 10})}> Inc </button>
<button type="button" onClick={() => dispatch({type:'DECREMENT'})}> DEC </button>
<button type="button" onClick={() => dispatch({type:'RESET'})}> RESET </button>
                                                                </div>
function counterReducer(state, action) {
   switch(action.type) {
      case 'INCREMENT':
        return {
                                                                                                                          count: 8
           count: state.count + action.payload
      case 'DECREMENT':
       return {
           count: state.count - 1
      case 'RESET':
        return {
           count: 0
                                                           function Sample() {
                                                              let [state, dispatch] = React.useReducer(counterReducer, {count:0});
                                                                crurn <alv>
    Count: {state.count} <br/>
    Count: {state.count} <br/>
    <button type="button" onClick={() => dispatch({type: 'INCREMENT', payload: 10})}> Inc </button>
    <button type="button" onClick={() => dispatch({type: 'DECREMENT'})}> DEC </button>
    <button type="button" onClick={() => dispatch({type: 'RESET'})}> RESET </button>
                                                               </div>
```