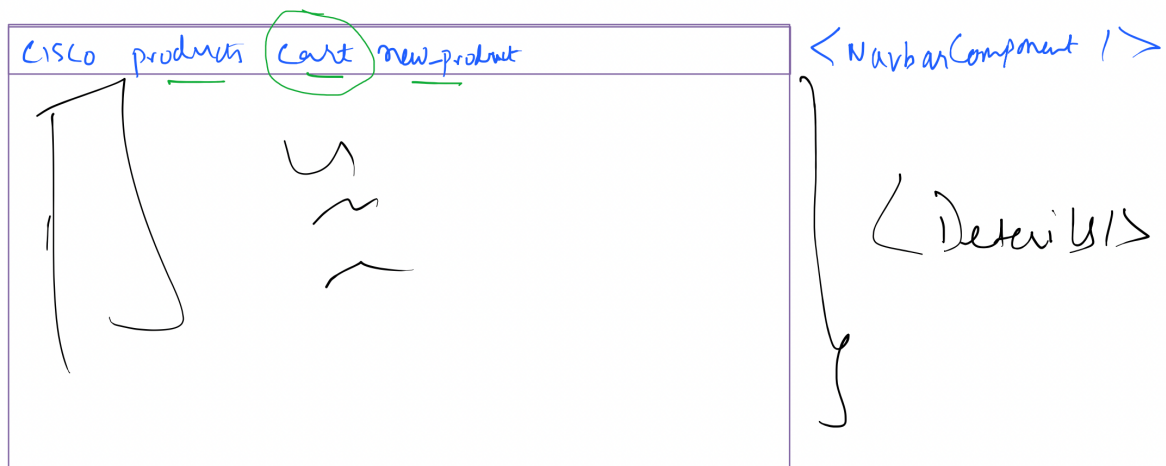
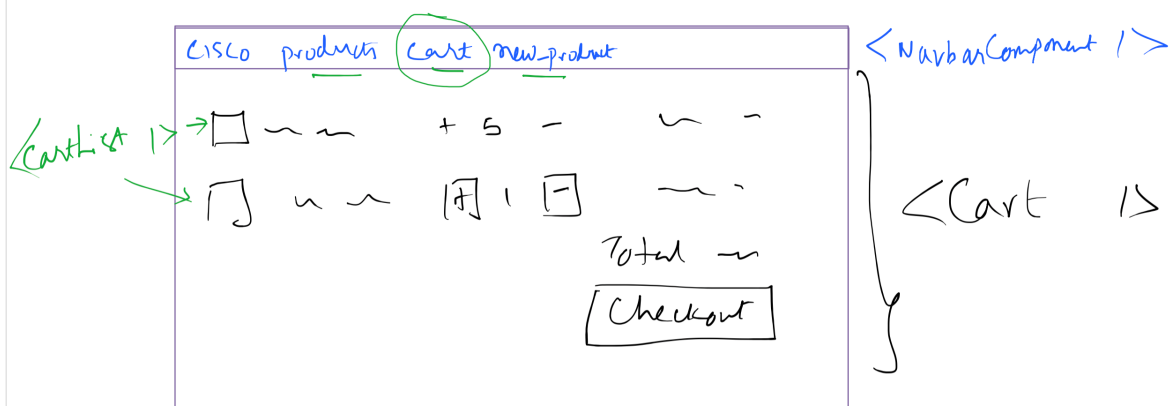
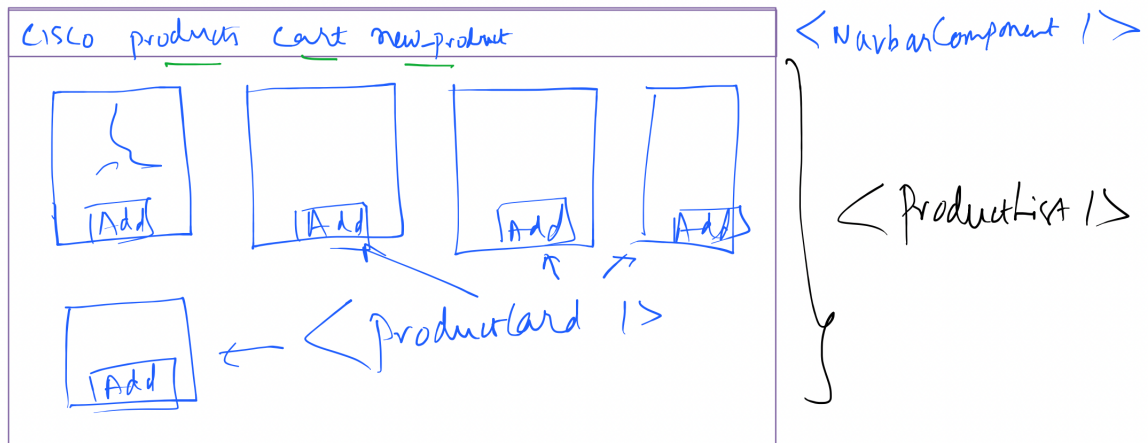


SPA



```
<CartContext.Provider value={{...state, addToCart, increment, clearCart}}>
```

```
export const CartContext = createContext();
```

```
cartItems: [],
total: 0,
quantity: 0
addToCart(item)
increment(id) clearCart()
```

```
<CartContext.Provider value={{...state, addToCart, increment, clearCart}}>
```

```
export const CartContext = createContext();
```

```

    cartItems: []
    total: 0,
    quantity: 0
    addToCart(item)
    increment(id)
    clearCart()
  
```

```

export default function ProductCard({ product }) {
  let { id, image, title, description, price } = product;
  let { addToCart } = useContext(CartContext);

```

```

<Button variant="primary" onClick={() => addToCart({
  id,
  image,
  title,
  description,
  price,
  quantity: 1,
  amount: price
})}>Add to Cart</Button>

```

```

function addToCart(item) {
  dispatch({type: 'ADD_TO_CART', payload: item});
}

```

```

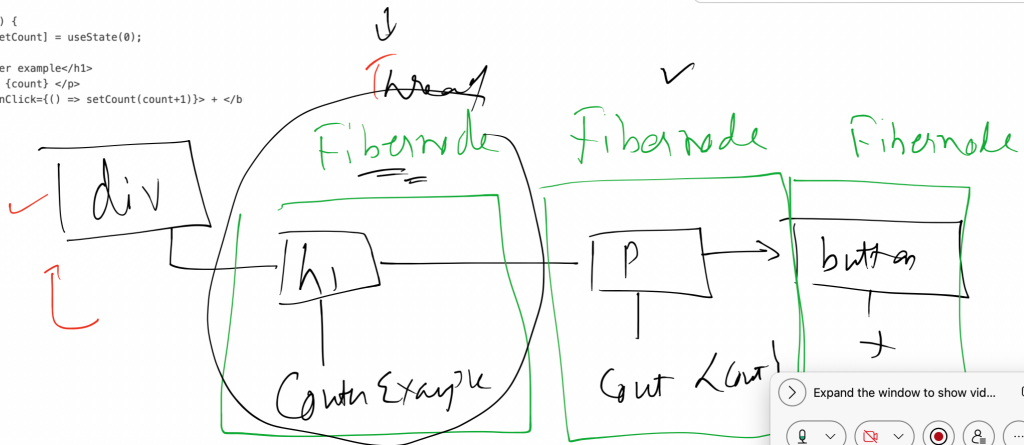
export default function cartReducer(state, action) {
  switch(action.type) {
    case 'ADD_TO_CART':
      return {
        cartItems: [...state.cartItems, action.payload],
        total: state.total + action.payload.price,
        quantity: state.quantity + 1
      };
  }
}

```

```

function Counter() {
  let [count, setCount] = useState(0);
  return <div>
    <h1>Counter example</h1>
    <p>Count {count}</p>
    <button onClick={() => setCount(count+1)}> + </b>
  </div>
}

```



Fork Join Algorithm

High Priority Thread = User Input