```
import java.util.concurrent.locks.ReentrantLock;
class BetterSafe implements State{
    private final ReentrantLock lock= new ReentrantLock();
    private byte maxval;
    private byte [] value;
    BetterSafe(byte[]v){
        value=v;
        maxval=127;
    }
    BetterSafe(byte[]v, byte m){
        value = v;
        maxval=m;
    }
    public int size(){
        return value.length;
    }
    public byte[] current(){
        return value;
    }
    public boolean swap(int i, int j){
        lock.lock();
        if ( value[i] <= 0 || value[j]>= maxval ) {
            lock.unlock();
            return false;
            }
            value[i]--;
            value[j]++;
            lock.unlock();
            return true;
    }
}
```