```
Input: data: array of positive integers
Input: n: size of data
Output: JubJub(data)

1  Algorithm: JubJubReduction
2  cap = max(data) // O(n) assuming linear search
3  for i = 1 to n // n iterations × [n + B(n)] = n² + nB(n)
4     x = min(data) // O(n) assuming linear search
5     if x > cap then // (O(1)
6        return false // O(1)
7     end
8     Bandersnatch(data) // O(B(n))
9  end
10 return true // O(1)
```

$j(n) \leq$ JubJubReduction $\leq J(n)$

$b(n) \leq$ Bandersnatch $\leq B(n)$

$n^{10} \leq B(n),\ b(n),\ J(n),\ j(n)$

1. **$O(n + n^2 + nB(n))$** or **$O(nB(n))$** since B(n) is $\Omega(n^{10})$

2.

    a. BS is O(J(n)/n) **FALSE**

    b. BS is $\Omega(j(n)/n)$ **TRUE**, since JJ $\leq n \times$ BS (#1), JJ/n $\leq$ BS $\rightarrow$ BS is $\Omega(j(n)/n)$

    c. JJ is O(nB(n)) **TRUE**, since JJ is O (nB(n))

    d. JJ is Ω(nb(n)) **FALSE**, since JJ is $\Omega(n)$