Solution for "Angry Cows" Bronze January 2016

By Riya Arora

For this problem, we have some haybales and a cow will be launched and hit a haybale which will explode and hit more haybales which then explode. Since there are only 100 haybales, we can try each haybale as the first one that explodes (the one that the cow hits). Once we pick the bale that the cow hits, we search to the left and separately to the right because when a bale explodes another bale, let's say to the left, the radius increases by 1 but the distance between the bales is at least one so the new bale will only explode new bales to the left.

In the sample input, we have 6 bales located at 8, 5, 6, 13, 3, 4. We try all the bales as our first bale that is hit, but let's just consider when the bale at location 6 is hit for now. The first thing we do is see how many bales from the left will explode. We can loop through all the bales to find the closest bale to the left which is 5. Since 5 is within distance 1 from 6, 5 explodes and we loop through finding the closest bale left of 5 which is 4. Since 4 is within 2 units of 5, 4 explodes and we loop through to find the closest bale left of 4 which is 3. Since 3 is within 3 units of 4, 3 explodes and we loop through to find the closest bale left of 3. Since there are no bales left of 3, we stop. We repeat this for the right side. The closest bale to the right of bale 6 is 8. Since 8 is not within 1 unit from 6, it doesn't explode and we stop. 4 bales exploded.

Here is the pseudocode:

```
answer = 0 /// initialize the answer to 0
read N
for i = 1...N
      read bales[i]

for i = 1...N
      /// i is the first bale that explodes
      good = true           /// good is if we are still exploding bales
      exploded = 0          //this is the count of how many bales exploded.
      current = bales[i]    /// the location of the current bale that is
exploding

      /// we are considering the explosion to the right in this while Loop
      add = 1               /// how far the bale will explode
      while good == true    /// while we are still making progress
            exploded++
            good = false    /// we will set it to true when we find the
            next bale
            next = 2000000000 /// set it to infinity
            for j = 1...N
                  if bales[j] > current and bales[j] < next and
                  bales[j] <= current + add
                        next = bales[j]
                        good = true /// we have found another bale
```

```
            current = next
            add++

      /// we now repeat this for exploding left
      good = true
      current = bales[i]
      add = 1
      while good == true    /// while we are still making progress
            exploded++
            good = false
            next = 0
            for j = 1...N
                  if bales[j] < current and bales[j] > next and
                  bales[j] >= current - add {
                        next = bales[j]
                        good = true

            current = next
            add += 1

      Exploded -= 1         /// we have counted the starting bale twice and we
      only want to count it once
      answer = max(answer, exploded)

output answer
```

To speed up the code and make it simpler to write, we can sort the array. Then the element to the left is the one with one index smaller and the element to the right is the one with one index bigger. We can use the same algorithm. The pseudocode is below:

```
answer = 0 /// initialize the answer to 0
read N
array bales[N]
read bales[]

sort bales

for i = 1...N
      /// i is the first bale that explodes
      exploded = 1          //this is the count of how many bales exploded.
      It is one because bale i has exploded
      j = i                 /// the index of the current bale that is exploding

      /// consider the explosion to the right in this while loop
      while j < n and bales[j+1] <= bales[j] + add
            exploded = exploded + 1
```

```
            add = add + 1
            j = j - 1

      /// we now repeat this for exploding left
      add = 1               /// how far the bale will explode
      j = i
      while j > 1 and bales[j-1] >= bales[j] - add
            exploded = exploded + 1
            add = add + 1
            j = j - 1

      answer = max(answer, exploded)

output answer
```