```
Greedy Magic (A[], i)
    n = length of (A[])
    max = A[i] · A[i]
    start = end = min = magic = sum = 0
    if  A[i+1] ≥ A[i-1]
        if i =! n
            j = i+1
        else
            j = i
            i = i-1
    else
        if i =! 0
            j = i
            i = i-1
        else
            i = i+1
    while (j-i) < n
        magic = minOf(A[i,...,j]) · sumOf(A[i,...,j])
        if magic > max
            max = magic
            start = i
            end = j
        (Check for either i+1 or i-1)
    (Final check of whole array since it stops at n-1)
    return [max, start, end]
```

Θ(n)

Since it starts with one element of A, then continues to loop through, adding one more element at a time, until the entire array has been checked. This means it loops through n-1 times, with constant time inside the loop making the algorithm Θ(n) time.