Homework #0

Raja Kantheti

August 29, 2024

1 Problem1

Determine whether all entries of array A are distinct, that is, return *True* if there are no duplicate entries in A, and *False* otherwise.

1.1 pseudocode

```
seen = {} # seen is a set, so no duplicate values.
for i in 0 to len(A)-1:
  do:
    if a[i] in seen:
      return False
    else:
      seen.append(a[i])
return True
```

1.2 Demonstration of Operation

Example1:
Example array A = [1,2,2,3,4]

The For loop starts at index 0 and does two things, checks if the value is already in the set *seen* and, if not, it adds the value at the current index to set *seen*. As the loop progresses, the append operation is executed for the elements at indices 0,1 to the set *seen*, which does not allow duplicates. After reaching the index 2 of the Array A, since the value at A[3], exists in the set *seen*, the function returns *False* and the loop is terminated.
Example2:
Example array A = [1,2,3,4,5]

The For loop starts at index 0 and appends the elements at indices 0,1,2,3,4 to the set *seen*, which does not allow duplicates. The index is exhausted, which means the Array is traversed once and there are no duplicates found. So, True is returned.

2 Problem2

Compute $\lfloor \sqrt{n} \rfloor$ for any positive integer n. Besides assignment and comparison, your algorithm may use only the four basic arithmetical operations $(+, -, *, /)$.

2.1 pseudocode

we can use a binary search for this.

```
high = n
low = 1
while low < high:
  mid = (low + high) / 2
  if mid * mid = n:
    return mid
  else if mid * mid < n:
    low = mid + 1
  else:
    high = mid - 1
return high
```

2.2 Demonstration of Operation

Example1:
For n = 9,
First Iteration:
low = 1
high = 9
mid = 5
5^2 is 25, which is greater than 9. So, high is updated to 4.
Second Iteration:
low = 1
high = 4
mid = 2
2^2 is 4, which is less than 9. So, low is updated to 3.

Third Iteration:
low = 3
high = 4
mid = 3
Since 3^2 is 9, it is returned.
Example2:
For n=4
First Iteration:
low =1
high =4
mid = 2
2^2 is 4, which is equal to n. So, 2 is returned.

3 Problem3

Compute the product mn for any positive integers m and n, using only addition (+).

3.1 pseudocode

```
function product(n, m):
  result = 0
  for i in range(0, m):
    result += n
  return result
```

3.2 Demonstration of Operation

Example1:
For m = 3, n = 4,
First Iteration:
result = 0
i = 0
result = 0 + 4 = 4
Second Iteration:
result = 4
i = 1
result = 4 + 4 = 8
Third Iteration:
result = 8

i = 2
result = 8 + 4 = 12
12 is returned.
Example2:
For m = 2, n = 5,
First Iteration:
result = 0
i = 0
result = 0 + 5 = 5
Second Iteration:
result = 5
i = 1
result = 5 + 5 = 10
10 is returned.

4 Problem4

Compute the quantity m^n for any positive integers m and n, using only addition (+) or your algorithm from Question 3.

4.1 pseudocode

```
function power(m, n):
  result = 1
  for i in range(0, n):
    result = product(result, m)
  return result
```

4.2 Demonstration of Operation

Example1:
For m = 2, n = 3,
First Iteration:
result = 1
i = 0
result = product(1, 2) = 2
Second Iteration:
result = 2
i = 1
result = product(2, 2) = 4

Third Iteration:
result = 4
i = 2
result = product(4, 2) = 8
8 is returned.
Example2:
For m = 3, n = 2,
First Iteration:
result = 1
i = 0
result = product(1, 3) = 3
Second Iteration:
result = 3
i = 1
result = product(3, 3) = 9
9 is returned.