

A: SRC=" Q: Make a recursive function work on numpy array Is there a way to implement a recursive function with Numpy arrays? I need to find the kth item in a list of n-length lists. Here is what I have so far:

```
def find_kth(data, k):
    if k == len(data):
        return data
    else:
        mid = (k - 1)//2
        res = data[:mid] + data[mid+1:]
        if len(res) > k:
            kth = data[:res[-1]]
        else:
            kth = data[:res[-1]]
        return kth, res
```

However, it's not working properly.

```
for i in range(5):
    print(find_kth([], i))
```

Results in (None, None) (None, None) (None, None) (None, None) (None, None)

It should return None when the list is empty, and when the list contains exactly k items.

A: def find_kth(data, k):
if k == len(data):
return data
mid = int(k / 2)
return find_kth(data[:mid] + data[mid+1:], k)

There are two issues with your code. One is that when you slice a numpy array, it returns a view, not a copy. Another is that you were overwriting your data list with the result of res, which will make all subsequent mutations to res affect data, which will all be overwritten again with the result of res.

Rumors make us think we know something that we don't know - mxfh

Download



 Download from
Dreamstime.com

 35103813
Yula Gapeenko | Dreamstime.com

