



Master algorithms together on [Binary Search](#)! Create a room, invite your friends, and race to finish the problems.

Daily Coding Problem

[Blog](#)

Daily Coding Problem #143

Problem

This problem was asked by Amazon.

Given a pivot x , and a list lst , partition the list into three parts.

- The first part contains all elements in lst that are less than x
- The second part contains all elements in lst that are equal to x
- The third part contains all elements in lst that are larger than x

Ordering within a part can be arbitrary.

For example, given $x = 10$ and $lst = [9, 12, 3, 5, 14, 10, 10]$, one partition may be $[9, 3, 5, 10, 10, 12, 14]$.

Solution

This question has a relatively simple $O(1)$ space and $O(n)$ time solution involving few passes.

- In the first pass, put all elements in $lst < x$ to the front
- In the second pass, put all elements in $lst > x$ to the end

One way to do it in one pass is to keep three variables, i , j , and k , with these invariants:

- All elements in $lst[:i]$ are less than x
- All elements in $lst[i:j]$ are equal to x
- All elements in $lst[k+1:]$ are greater than x

Then we iterate with j and put $lst[j]$ according to the above invariants.

```
def partition(lst, x):
    i = 0
    j = 0
    k = len(lst) - 1

    while j < k:
        if lst[j] == x:
            j += 1
        elif lst[j] < x:
            lst[i], lst[j] = lst[j], lst[i]
            i += 1
            j += 1
        else:
            lst[j], lst[k] = lst[k], lst[j]
            k -= 1

    return lst
```

This will take only $O(1)$ space and $O(n)$ time.

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)