

Python 3.12.0 (v3.12.0:0fb18b02c8, Oct 2 2023, 09:45:56) [Clang 13.0.0 (clang-1300.0.29.30)]
on darwin
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: /Users/jreid/Documents/JLR_dev_code/merrimack/CSC6013/E6.py =====

Jesse Reid CSC6013 In Class Exercise E6

Russian Peasants Multiplication:

Multiplying 64 and 13 using the Russian Peasant method

Recursively dividing by 64 and multiplying by 13

| 64 | 13 |

| /2 | *2 |

32	26
16	52
8	104
4	208
2	416
1	832

Checking the odd numbers in the division list

| 64 | 13 | |

| /2 | *2 |

32	26	
16	52	
8	104	
4	208	
2	416	
1	832	√

After adding the checked values
64 x 13 is 832

Multiplying 60 and 13 using the Russian Peasant method

Recursively dividing by 60 and multiplying by 13

60 13
/2 *2

30 26
15 52
7 104
3 208
1 416

Checking the odd numbers in the division list

60 13
/2 *2

30 26
15 52 √
7 104 √
3 208 √
1 416 √

After adding the checked values
60 x 13 is 780

Multiplying 59 and 13 using the Russian Peasant method

Recursively dividing by 59 and multiplying by 13

59 13
/2 *2

29 26
14 52
7 104
3 208
1 416

Checking the odd numbers in the division list

59 13 √
/2 *2

29	26	√
14	52	
7	104	√
3	208	√
1	416	√

After adding the checked values
59 x 13 is 767

Lomuto partition:

Starting array = [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85] with 85 as the pivot.

Current array [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85]
 Elements less than pivot []
 Elements greater than pivot []
 Elements not visited [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85]

j = 100, i = 100
 100 > 85; no swap
 Current array [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85]
 Elements less than pivot []
 Elements greater than pivot [100]
 Elements not visited [33, 22, 213, 65, 29, 153, 199, 47, 181, 85]

j = 33, i = 100
 33 < 85; swapping 33 with 100
 Current array [33, 100, 22, 213, 65, 29, 153, 199, 47, 181, 85]
 Elements less than pivot [33]
 Elements greater than pivot [100]
 Elements not visited [22, 213, 65, 29, 153, 199, 47, 181, 85]

j = 22, i = 100
 22 < 85; swapping 22 with 100
 Current array [33, 22, 100, 213, 65, 29, 153, 199, 47, 181, 85]
 Elements less than pivot [33, 22]
 Elements greater than pivot [100]
 Elements not visited [213, 65, 29, 153, 199, 47, 181, 85]

j = 213, i = 100
 213 > 85; no swap
 Current array [33, 22, 100, 213, 65, 29, 153, 199, 47, 181, 85]
 Elements less than pivot [33, 22]
 Elements greater than pivot [100, 213]
 Elements not visited [65, 29, 153, 199, 47, 181, 85]

j = 65, i = 100
 65 < 85; swapping 65 with 100
 Current array [33, 22, 65, 213, 100, 29, 153, 199, 47, 181, 85]

Elements less than pivot [33, 22, 65]
Elements greater than pivot [213, 100]
Elements not visited [29, 153, 199, 47, 181, 85]

j = 29, i = 213
29 < 85; swapping 29 with 213
Current array [33, 22, 65, 29, 100, 213, 153, 199, 47, 181, 85]
Elements less than pivot [33, 22, 65, 29]
Elements greater than pivot [100, 213]
Elements not visited [153, 199, 47, 181, 85]

j = 153, i = 100
153 > 85; no swap
Current array [33, 22, 65, 29, 100, 213, 153, 199, 47, 181, 85]
Elements less than pivot [33, 22, 65, 29]
Elements greater than pivot [100, 213, 153]
Elements not visited [199, 47, 181, 85]

j = 199, i = 100
199 > 85; no swap
Current array [33, 22, 65, 29, 100, 213, 153, 199, 47, 181, 85]
Elements less than pivot [33, 22, 65, 29]
Elements greater than pivot [100, 213, 153, 199]
Elements not visited [47, 181, 85]

j = 47, i = 100
47 < 85; swapping 47 with 100
Current array [33, 22, 65, 29, 47, 213, 153, 199, 100, 181, 85]
Elements less than pivot [33, 22, 65, 29, 47]
Elements greater than pivot [213, 153, 199, 100]
Elements not visited [181, 85]

j = 181, i = 213
181 > 85; no swap
Final swap 213 for 85

Lomuto with pivot at 5 : 85
Final array [33, 22, 65, 29, 47, 85, 153, 199, 100, 181, 213]

Items less than pivot [33, 22, 65, 29, 47]
Items greater than pivot [153, 199, 100, 181, 213]

Code below:

Jesse Reid CSC6013 In Class Exercise E6

1. TracetheRussianPeasantsMultiplicationalgorithmforthe following products. Show each recursive call and the final result, as shown in the live session (table).

a. 64×13 b. 60×13 c. 59×13

function to recursively obtain the values needed for Russian Peasant multiplication

```
def russian(n, m):
```

```
    div = []
```

```
    mul = []
```

```
    peasant_sum = 0
```

```
    dcount = 0
```

```
    ecount = 0
```

```
    print(f"\nRecursively dividing by {n} and multiplying by {m}")
```

```
    # recursively divide all "n" values by 2
```

```
    def division(n):
```

```
        if n == 1:
```

```
            return div
```

```
        else:
```

```
            n = int(n/2)
```

```
            div.append(n)
```

```
            return division(n)
```

```
    # recursively multiply all "m" values by 2
```

```
    def multiply(m, count):
```

```
        if count == 0:
```

```
            return mul
```

```
        else:
```

```
            count -= 1
```

```
            m = int(m * 2)
```

```
            mul.append(m)
```

```
            return multiply(m, count)
```

```
    # call the recursive functions
```

```
    div = division(n)
```

```
    count = len(div)
```

```
    mul = multiply(m, count)
```

```
    # create the first table
```

```
    print(f"\n| {n} | {m} | \n_____ \n| /2 | *2 \n_____ \n")
```

```
    for i in div:
```

```
        print(f"| {i} | {mul[dcount]} |")
```

```
        dcount += 1
```

```
    print("_____ \n")
```

```
    print("\nChecking the odd numbers in the division list")
```

```
    # create the second table, checking all odd numbers and adding them
```

```
    if n % 2 != 0:
```

```
        print(f"\n| {n} | {m} | √ \n_____ \n| /2 | *2 \n_____ \n")
```

```
        peasant_sum += m
```

```
    else:
```

```
        print(f"\n| {n} | {m} | \n_____ \n| /2 | *2 \n_____ \n")
```

```
    for e in div:
```

```
        if e % 2 != 0:
```

```
            print(f"| {e} | {mul[ecount]} | √ |")
```

```

    peasant_sum += mul[ecount]
else:
    print(f" {e} | {mul[ecount]} |  |")

    ecount += 1
print("_____\\n")

print(f"After adding the checked values\\n{n} x {m} is {peasant_sum}\\n\\n")

```

Lomuto partition
2. TracetheLomutopartitionwiththearray:
a. A=[100,33,22,213,65,29,153,199,47,181,85]
Using A[10] = 85 as pivot the final array will be:
• A = [33, 22, 65, 29, 47, 85, 153, 199, 100, 181, 213]
In your trace, write down to each change in either i or j, stating: the values of i and j, swaps made,
and elements divided into lesser than the pivot, greater than the pivot, and yet to compare.

```

# recursive function to execute the Lomuto partition
def lomuto(A, left, right):
    pivot = A[10]
    # print the stating array and pivot
    print(f"Starting array = {A} with {pivot} as the pivot.\\n")
    p = A[right]
    i = left
    # itirate through A
    for j in range(left, right):
        # print the current status of the array and visited values
        print(f"Current array {A}")
        print(f"Elements less than pivot {A[:i]}")
        print(f"Elements greater than pivot {A[i:j]}")
        print(f"Elements not visited {A[j:]}\\n")

        # print the current values for j and i
        print(f"j = {A[j]}, i = {A[i]}")

        if A[j] < p:
            # indicate that a swap was made
            print(f"{A[j]} < {p}; swapping {A[j]} with {A[i]}")
            A[i], A[j] = A[j], A[i]

            i += 1

        else:
            # indicate that no swap was made
            print(f"{A[j]} > {p}; no swap")

    # final swap for the pivot itself
    A[i], A[right] = A[right], A[i]
    print(f"Final swap {A[right]} for {A[i]}\\n")

    return i

```

```

def main():
    print("Jesse Reid CSC6013 In Class Exercise E6\n\n")
    print("_____ \nRussian Peasants Multiplication:
\n_____ \n")
    values = {64:13, 60:13, 59:13}
    for n , m in values.items():
        print(f"Multiplying {n} and {m} using the Russian Peasant method")
        russian(n, m)

    print("_____ \nLomuto partition:\n_____ \n")
    A = [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85]
    pvt = lomuto(A, 0, len(A)-1)
    print("Lomuto with pivot at", pvt, ":", A[pvt])
    print(f"Final array {A}")
    print(f"\nItems less than pivot {A[:pvt]}")
    print(f"Items greater than pivot {A[pvt+1:]}")

if __name__ == "__main__":
    main()

```