

Chapter 6 Answers to some of questions

True/False Questions

- 1) False (recursive functions call themselves, there are quite useful ones)
- 2) True (page 212, Summary)
- 3) False (recursive functions must have at least one base case, i.e. it might have 1 or 2 or 3..)
- 6) True

$\overline{1^{\text{st}} \text{ letter}}$ $\overline{2^{\text{nd}} \text{ letter}}$ $\overline{3^{\text{rd}} \text{ letter}}$... $\overline{(n-1)^{\text{th}} \text{ letter}}$ $\overline{n^{\text{th}} \text{ letter}}$
n choices (n-1) choices (n-2) choices ... n-((n-1) -1) = 2 choices n-(n-1) = 1 choice
(*"choices" stands for "letters to choose from"*)
i.e. we get $n \times (n-1) \times (n-2) \times \dots \times 2 \times 1 = n!$

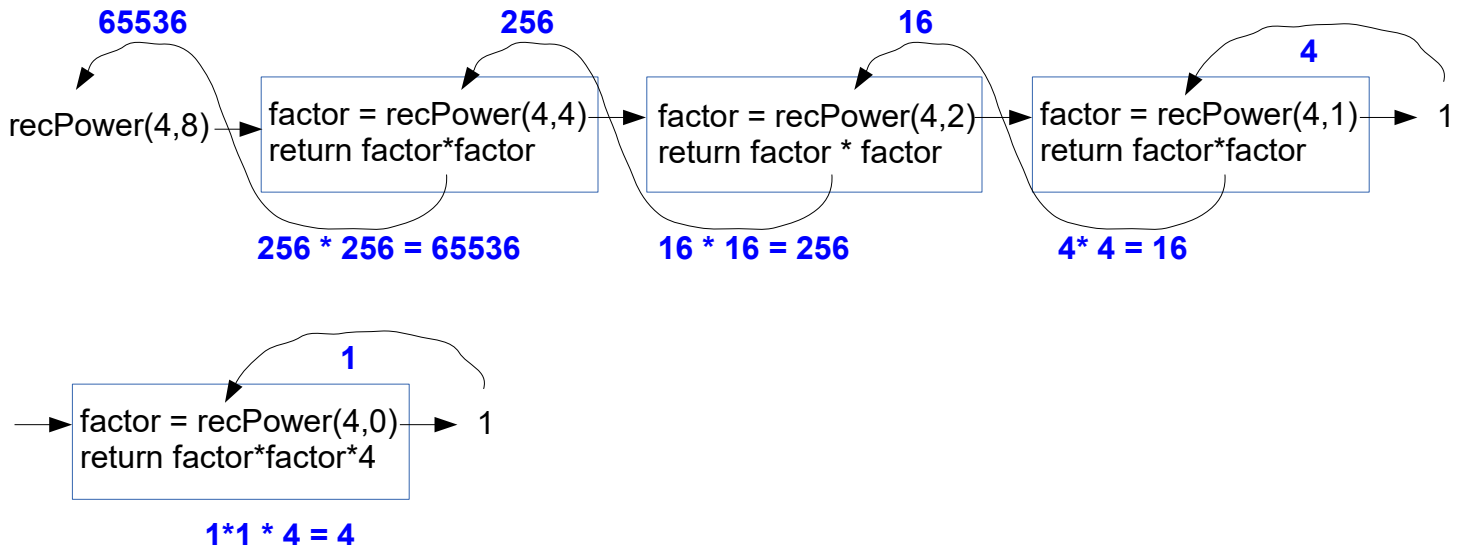
Multiple Choice questions

- 1) d)
- 4) d) (because $4! = 1*2*3*4 = 24$)
- 5) c) (n iterations)
- 6) a) (the power is divided by two which gives $\log n$ recursive calls)

Short-Answer questions

- 1) Yes. A proper recursive function must have at least one base case, which will be checked by a decision structure (control structure, like `if ...`)

Trace `recPower(4,8)` and figure out exactly how many multiplications it does.



Chapter 6 homework solutions and answers

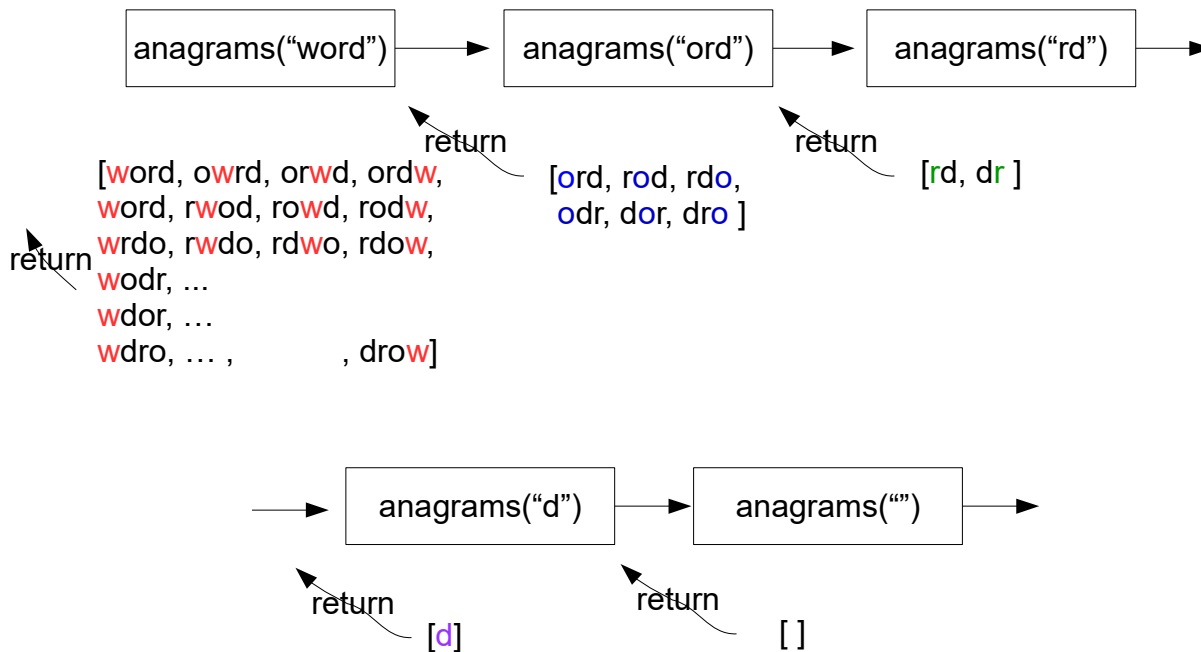
1) What list is returned by `anagrams("word")`?

Preliminary analysis:

First of all, we expect $4! = 24$ of them ($\frac{4}{4} \frac{3}{3} \frac{2}{2} \frac{1}{1}$) - number of choices

Secondly, we can simply run the program and see the result, but what's most important is to understand how that result was achieved!

Pictorial representation of `anagram("word")` call:



Result: `['word', 'owrd', 'orwd', 'ordw', 'wrod', 'rwod', 'rowd', 'rodw', 'wrdo', 'rwdo', 'rdwo', 'rdow', 'wodr', 'owdr', 'odwr', 'odrw', 'wdor', 'dwor', 'dowr', 'dorw', 'wdro', 'dwro', 'drwo', 'drow']`

If you add `print(ans)` right before `return ans`, this is what you will see:

```

['d']
['rd', 'dr']
['ord', 'rod', 'rdo', 'odr', 'dor', 'dro']
['word', 'owrd', 'orwd', 'ordw', 'wrod', 'rwod', 'rowd', 'rodw', 'wrdo', 'rwdo', 'rdwo', 'rdow', 'wodr', 'owdr',
'odwr', 'odrw', 'wdor', 'dwor', 'dowr', 'dorw', 'wdro', 'dwro', 'drwo', 'drow']
['word', 'owrd', 'orwd', 'ordw', 'wrod', 'rwod', 'rowd', 'rodw', 'wrdo', 'rwdo', 'rdwo', 'rdow', 'wodr', 'owdr',
'odwr', 'odrw', 'wdor', 'dwor', 'dowr', 'dorw', 'wdro', 'dwro', 'drwo', 'drow']
  
```