We are recovering from significant hosting issues. Much of the site is functional, but currently email delivery is not. Please bear with us as we validate site functionality.

Bookmarks

- Welcome to the edX Platform
- Entrance Survey
- Download Python and Get Motivated!
- Week 1: Python Basics
- Week 2: Simple Programs
- Week 3: Structured Types
- Week 4: Good Programming Practices
- Midterm Exam
- Week 5: Object Oriented Programming

Week 6: Algorithmic Complexity > 11. Computational Complexity > Exercise 6

Exercise 6

Bookmark this page

Exercise 6

4 points possible (graded)

ESTIMATED TIME TO COMPLETE: 6 minutes

Consider the following Python procedures. For each one, specify its order of growth.

1.

```
# Assume n has been previously bound to some value
i = 0
while i < 5:
   n *= 2
   i += 1

print(n)
```

Select an option

2.

```
def iterPower(a, b):
   result = 1
   while b > 0:
      result *= a
      b -= 1
   return result
```

Select an option

3.

```
def recurPower(a, b):
   print(a, b)
   if b == 0:
      return 1
   else:
      return a * recurPower(a, b-1)
```

- Week 6: Algorithmic Complexity
  - 11. Computational Complexity
    Finger Exercises
  - 12. Searching and Sorting Algorithms
    Finger Exercises
  - Problem Set 6
    Problem Set due Mar 9, 2017 15:30 PST
- Week 7: Plotting
- Exit Survey
- Sandbox

Select an option

4.

```
def recurPowerNew(a, b):
   print(a, b)
   if b == 0:
      return 1
   elif b%2 == 0:
      return recurPowerNew(a*a, b/2)
   else:
      return a * recurPowerNew(a, b-1)
```

Select an option

Submit

Exercise 6

Topic: Lecture 11 / Exercise 6

Show Discussion