소프트웨어 프로젝트 2

1조
정진혁, 조가성, 조경상, 조상연

Cord Review
(완성된 코드(조경상))

ChoKyungSang 조경상-recursivefactorial 11f9db3 4 days ago

1 contributor

18 lines (12 sloc) | 252 Bytes

Raw Blame History

```
1  def recursivefactorial(x):
2          if x == 1 or x == 0:
3                  return 1
4          else :
5                  a = recursivefactorial(x-1) * x
6                  return a
7
8
9  while True:
10
11         n = int(input("수를 입력하세요: "))
12         if n <= -1:
13                 break
14         answer = recursivefactorial(n)
15
16         print(n,"! = ",answer)
17
```

Cord Review
(코드 실행(조경상))

조경상 코드!.py - C:₩Users₩조상연₩Documents₩python₩조경상 코드

File Edit Format Run Options Window Help

```
def recursivefactorial(x):
        if x == 1 or x == 0:
                return 1
        else :
                a = recursivefactorial(x-1) * x
                return a


while True:

        n = int(input("수를 입력하세요: "))
        if n <= -1:
                break
        answer = recursivefactorial(n)

        print(n,"! = ",answer)
```

Python 3.6.1 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:₩Users₩조상연₩Documents₩python₩조경상 코드!.py ==========
수를 입력하세요: 3
3 ! =  6
수를 입력하세요: 6
6 ! =  720
수를 입력하세요: 12
12 ! =  479001600
수를 입력하세요: 20
20 ! =  2432902008176640000
수를 입력하세요: 100
100 ! =  933262154439441526816992388562667004907159682643816214685929638952175999932299156089
414639761565182862536979208272237582511852109168640000000000000000000000000
수를 입력하세요: -1
>>>
```

Ln: 16 Col: 4

Cord Review
(완성된 코드(조상연))

KingGodGeneralJSY homework4 commit by using time module

0 contributors

1

34 lines (24 sloc) | 703 Bytes

```python
import time                 #time모듈로 재귀함수와 for loop의 차이를 알아보자!

def recursive_factorial(n):     #재귀함수로 짜여진 팩토리얼 계산
    if n==0:
        return '0!=1'

    if n == 1:
        return 1
    else:
        return recursive_factorial(n-1)*n
```

2

```python
def for_loop_factorial(n):      #for loop으로 짜여진 팩토리얼 계산
    if n==0:
        return '0!=1'

    fac_num = n
    for i in range(1,n):
        fac_num = fac_num*i

    return fac_num
```

3

```python
#time 모듈로 둘의 차이를 알아보자!
ts=time.time()
print(recursive_factorial(10))
ts= time.time() - ts
print(ts)

ts=time.time()
print(for_loop_factorial(10))
ts= time.time() - ts
print(ts)
```

Cord Review
(코드 실행(조상연))

Python 3.6.1 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/조상연/Documents/python/assignment4_20171706.py ======
=
3628800
0.0050394535064697266
3628800
0.0010023117065429688
>>>
======= RESTART: C:/Users/조상연/Documents/python/ass
=
9332621544394415268169923885626670049071596826438162
6089414639761565182862536979208272237582511852109168
0.012965440750122 07
9332621544394415268169923885626670049071596826438162
6089414639761565182862536979208272237582511852109168
0.004010200500488281
>>>
```

N=10일 때

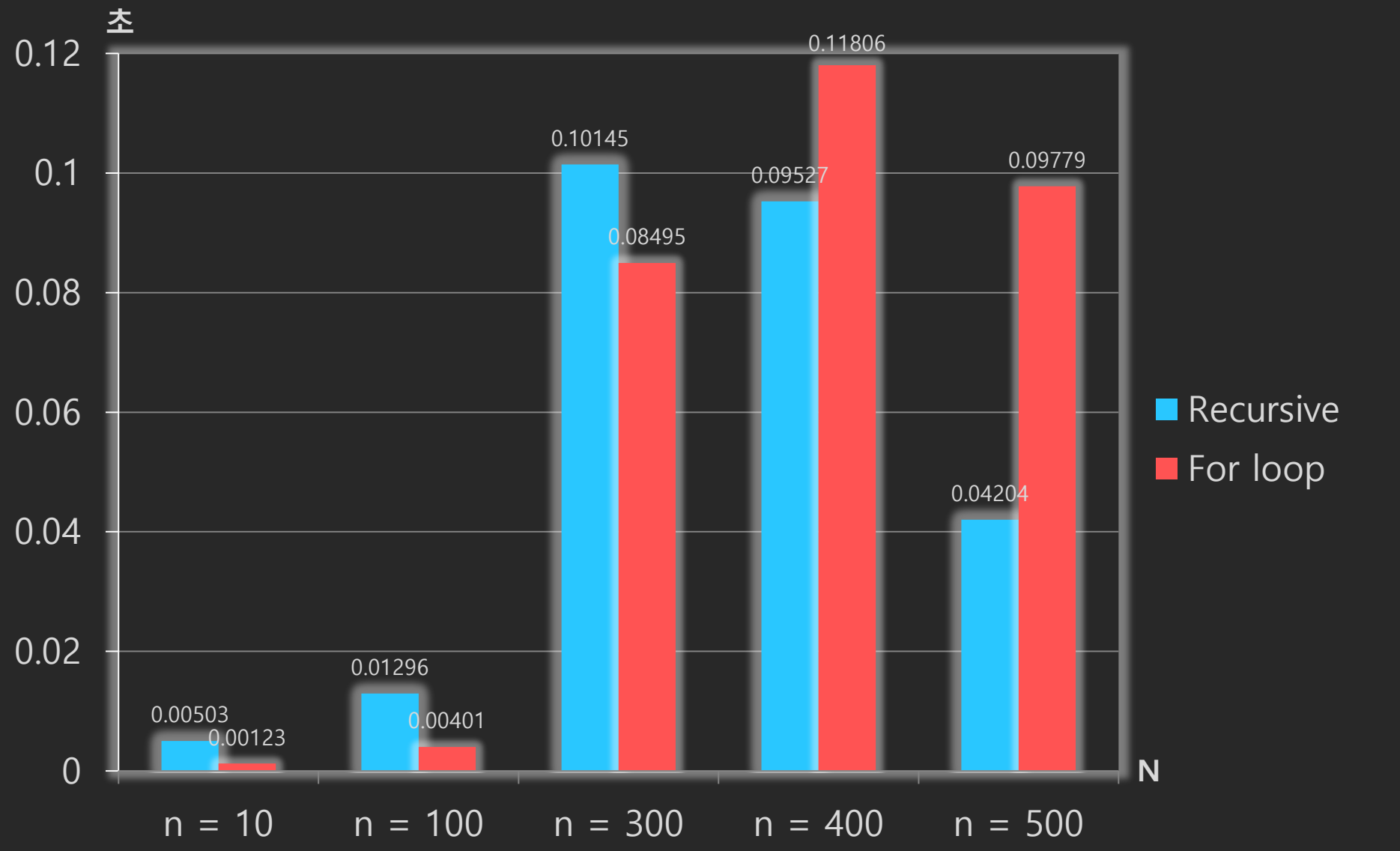
N=100일 때

N=500일 때 ⟶

```
======= RESTART: C:/Users/조상연/Documents/python/assignment4_20171706.py ======
=
1220136825991110068701238785423046926253574342803192842192413588385845373153881
9976054964475022032818630136164771482035841633787220781772004807852051593292854
7790757193933060377296085908627042917454788242491272634430567017327076946106280
2310452644218878789465754777149863494367781037644274033827365397471386477878495
4384895955375379904232410612713269843277457155463099772027810145610811883737095
3101635632443298702956389662891165897476957208792692887128178007026517450776841
0719624390394322536422605234945850129918571501248706961568141625359056693423813
0088562492468915641267756544818865065938479517753608940057452389403357984763639
4490531062323749066445048824665075946735862074637925184200459369692981022263971
9525971909452178233317569345815085523328207628200234026269078983424517120062077
1464097945611612762914595123722991334016955236385094288559201872743379517301458
6357570828355780158735432768888680120399882384702151467605445407663535984174430
4801289383138968816394874696588175045069263653381750554781286400000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000
0.04204511642456055
1220136825991110068701238785423046926253574342803192842192413588385845373153881
9976054964475022032818630136164771482035841633787220781772004807852051593292854
7790757193933060377296085908627042917454788242491272634430567017327076946106280
2310452644218878789465754777149863494367781037644274033827365397471386477878495
4384895955375379904232410612713269843277457155463099772027810145610811883737095
3101635632443298702956389662891165897476957208792692887128178007026517450776841
0719624390394322536422605234945850129918571501248706961568141625359056693423813
0088562492468915641267756544818865065938479517753608940057452389403357984763639
4490531062323749066445048824665075946735862074637925184200459369692981022263971
9525971909452178233317569345815085523328207628200234026269078983424517120062077
1464097945611612762914595123722991334016955236385094288559201872743379517301458
6357570828355780158735432768888680120399882384702151467605445407663535984174430
4801289383138968816394874696588175045069263653381750554781286400000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000
0.09779191017150879
>>>
```

Cord Review
(결과 비교)

결과 정리



Cord Review
(결과)

결 ! 론

숫자가 작을 때에는 Recursive 보다 For loop이 효율적임
반면에 숫자가 점점 커질 때에는 Recursive가 For loop보다 효율적임

N이 작을 때, Recursive < For loop
N이 클 때, Recursive > For loop

감사합니다!