CPE231 Algorithm Design

Group ____________________________

Student ID ____________________________

Part I) From these simple codes, determine their asymptotic running time

1) nums is a list of size *n*

```
nums.append(1)
```

---

2) nums is a list of size *n*

```
nums.insert(0,2)
```

---

3) seq is a list with *n* elements

```
s = 0
for x in seq:
    s += x
```

---

4) seq is a list with *n* elements

```
squares = [x**2 for x in seq]
```

---

5) seq is a list with *n* elements

```
s = 0
for x in seq:
    for y in seq:
        s += x*y
```

---

6) seq is a list with *n* elements

```
s = 0
for x in seq:
    for y in seq:
        s += x*y
    for z in seq:
        for w in seq:
            s += x-w
```

---

7) seq1 contains n elements and seq2 contains m elements

```
s = 0
for x in seq1:
    for y in seq2:
        s+= x*y
```

---

8)
```
seq1 = [[0,1],[2],[3,4,5]]
    s = 0
    for seq2 in seq1:
        for x in seq2:
            s += x
```

---

9) seq is a list with *n* elements

```
    s = 0
    n = len(seq)
    for i in range(n-1):
        for j in range(i+1, n):
            s += seq[i] * seq[j]
```

---

10) seq is a list with *n* elements

```
def sort_w_check(seq):
    n = len(seq)
    for i in range(n-1):
        if seq[i] > seq[i+1]:
            break
    else:
        return
    …

>>> sort_w_check(seq)
```

Part II) Write code and do some experiments for these problem

1) Run a timing experiment for the recursive and non-recursive algorithm for computing n! and discuss with the theorical estimation

---

2) Run a timing experiment for the recursive and non-recursive algorithm for computing Fibonacci(n) and discuss with the theorical estimation

---