INT202 Complexity of Algorithms

Tutorial 2

Q1. Compute the following expressions that are given in postfix notation.

(a) 1 2 + 4 * 3 +

(b) 5 7 * 3 4 1 + * -

(c) 3 4 2 * 1 5 - 2 ^ / +

Note: The ^ symbol means exponentiation, i.e. x y ^ means x^y in the usual infix notation.

Q2. Compute Big-Oh notation.

a)
```
void func(int n){
  int i = 0;
  while(i * i * i <= n)
    i++;
}
```

b)
```
void recursive(int n, int m, int o) {
  if (n <= 0) {
    printf("%d, %d\n", m, o);
  } else {
    recursive(n - 1, m + 1, o);
    recursive(n - 1, m, o + 1);
  }
}
```

c)
```
void func(int n) {
  for (int i = 1, s = 0; i <= n; ++i) {
        int t = 1;
        for (int j = 1; j <= i; ++j)
              t = t * j;
  }
}
```

Q3. State whether the following statement is true or false, and give a brief justification for your answer.

a) If $f(n)$ and $g(n)$ are both $O(h(n))$, then $f(n)+g(n)$ is $O(h(n))$.

b) The asymptotic complexity of $\sum_{i=1}^{n} \log i$ is $O(\log n)$

Q4. Consider the following code fragment.

```
for i=1 to n do
        for j=i to 2*i do
                output hello
```

Let T(n) denote the number of times 'hello' is printed as a function of n.

1) Express T(n) as a summation;

2) Simplify the summation and give the worst-case running time using Big-Oh notation.

Q5. Consider the following code fragment.

```
for i=1 to n-1 do
        for j=i+1 to n do
                for k=1 to j do
                        output hello
```

Let T(n) denote the number of times 'hello' is printed as a function of n.

1) Express T(n) as a summation;

2) Simplify the summation, and give the worst-case running time using Big-Oh notation.

Q6. Give a tight bound of the runtime complexity class for the following code fragment in Big-Oh notation, in terms of the variable N.

```
int sum = N;
        for (int i = 0; i < 1000; i++) {
        for (int j = 1; j <= i; j++) {
        sum += N;}
        for (int j = 1; j <=i; j++) {
        sum += N;}
        for (int j = 1; j <= i; j++) {
        sum += N;}
}
```

Q7. Prove that $3logn + loglogn$ is $\Theta(logn)$