

## SC1007 Data Structures and Algorithms

### Tutorial 4: Analysis of Algorithm

**Q1** The function `subset()` below takes two linked lists of integers and determines whether the first is a subset of the second. Assume there are no duplicate numbers in each list.

- (a) When will the worst case happen?
- (b) In the worst case, how many item comparison operations will be made?
- (c) Give the worst-case running time of `subset` as a function of the lengths of the two lists.
- (d) Write down the worst-case running time complexity in asymptotic notations in terms of the lengths of the two lists.

```
1  # Check whether integer X is an element of linked list Q
2  def element(X, Q):
3      found = False # Flag whether X has been found
4      while Q is not None and not found:
5          found = (Q.item == X)
6          Q = Q.next
7      return found
8
9  # Check whether L is a subset of M
10 def subset(L, M):
11     success = True # Flag whether L is a subset so far
12     while L is not None and success:
13         success = element(L.item, M)
14         L = L.next
15     return success
```

4-1

**Q2** Find the number of `printf` called in the following functions. Write down its time complexity in  $\Theta$  notation in terms of  $N$ .

**Hint:** For repetition structure, compute the number of inner loops and outer loops. For recursive structure, write down the number of `printf`  $W(N)$  in terms of  $W(N-1)$ . Then use backward substitutions or forward substitutions to express  $W(N)$  in terms of  $N$ . You may refer to the series shown in lecture.

```

1  def Q2a(N):
2      j = 1
3      while j <= N:
4          k = 1
5          while k <= N:
6              print("SC 1007")
7              k *= 2
8          j *= 3
9
10 def Q2b(N):
11     if N > 0:
12         for i in range(N):
13             print("SC 1007")
14         Q2b(N - 1)
15         Q2b(N - 1)

```

4-2

**Q3** A sequence,  $x_1, x_2, \dots, x_n$ , is said to be cyclically sorted if the smallest number in the sequence is  $x_i$  for some  $i$ , and the sequence,  $x_i, x_{i+1}, \dots, x_n, x_1, x_2, \dots, x_{i-1}$  is sorted in increasing order. Design an algorithm to find the minimal element in the sequence in  $O(\log n)$  time. What is the worst-case scenario?