3.

For this algorithm I would implement a stack. Using a scanner to read the input, and using another scanner to ready every character in the input. I would push all of the characters on the stack until it comes to a closing parenthesis. Then I would pop all of the characters off up until the last opening parenthesis. Then I would push until the next closing parentheses, and so on. For every closing parenthesis it would add a new line.

It would look like this…

Input: (Go ((cake) is) od)

[ ( Go ( ( cake ]

output : cake

[ ( Go ( is ]

output: cake

is

[ ( Good ]

output: cake

is

Good

[ ]

The growth rate would be O(n^2)