Lab13-Algorithm

CS101-计算机科学导论课后作业，讲师：John Hopcroft，2016 秋季学期

* 可打印本页直接在页面作答/附纸装订。标注您的 姓名：____________ 学号：___________ 班级：______

* 电子版作业请直接上传到课程网站

Solve 3 of the following exercises.

1. Lexer and Parser are important components of a Compiler. A lexer converts the source code (a sequence of characters) into a sequence of tokens (a token could be generated by a *Regular Expression*), and a parser converts a sequence of tokens into a parse tree (which could be generated by a *Context Free Grammar*).

   Given the RE description of the tokens and the CFG of the programming language:

   - Tokens:

   $$\begin{aligned} assign &: \leftarrow \\ left\_bracket &: (\\ right\_bracket &:) \\ operator &: +|- \\ number &: [0-9]^+ \\ identifier &: [a-zA-Z0-9]^+ \end{aligned}$$

   - CFG:

   $$\begin{aligned} \textbf{PROGRAM} &: \textbf{STATEMENT}^+ \\ \textbf{STATEMENT} &: identifier\ assign\ \textbf{EXPRESSION} \\ \textbf{EXPRESSION} &: identifier \\ &|\ number \\ &|\ left\_bracket\ \textbf{EXPRESSION}\ right\_bracket \\ &|\ \textbf{EXPRESSION}\ operator\ \textbf{EXPRESSION} \end{aligned}$$

   And you need to use Dynamic Programming to parse the following program:

   $$\begin{aligned} a &\leftarrow 1 \\ b &\leftarrow a-(a+a) \end{aligned}$$

   You only need to write down the table used in DP.

   HINT: Firstly, you need to convert the CFG to a *Chomsky Normal Form* one.

2. Construct a *divide and conquer* algorithm to rearrange the sequence $1, 2, 3 \cdots, n$ that satisfies there is no arithmetic subsequence with length ≥ 3.

3. Prove the *Master theorem*.

   Master Theorem: Let a be an integer greater than or equal to 1 and b be a real number greater than 1. Let c be a positive real number and d a nonnegative real number. Given a recurrence of the form

   $$T(n) = \begin{cases} aT(n/b) + n^c & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases}$$

   then for n a power of b,

(a) if $\log_b a < c, T(n) = \Theta(n^c)$,
(b) if $\log_b a = c, T(n) = \Theta(n^c \log n)$,
(c) if $\log_b a > c, T(n) = \Theta(n^{\log_b a})$.

4. It is possible to perform multiplication of large numbers in (many) fewer operations than the usual brute-force technique of "long multiplication". As discovered by Karatsuba in 1962, multiplication of two n-digit numbers num_1 and num_2 can be done with a complexity less then $O(n^2)$ using identities of the form

$$A = (a + b \cdot 10^{\lceil n/2 \rceil}) \text{ and } B = (c + d \cdot 10^{\lceil n/2 \rceil}).$$

Then

$$A \cdot B = ac + [(a+b)(c+d) - ac - bd]] \cdot 10^{\lceil n/2 \rceil} + bd \cdot 10^n.$$

Use the *Master Theorem* to give a bound of *Karatsuba Algorithm*.

5. Prove the *Cut lemma*:

Cut Lemma: Let $X \subseteq T$ where T is an MST (Minimum Spanning Tree) in $G(V, E)$. Let $S \subseteq V$ such that no edge in X crosses between S and $V - S$; i.e., no edge in X has one endpoint in S and one endpoint in $V - S$. Among edges crossing between S and $V - S$, let e be an edge of minimum weight. Then $X \cup \{e\} \subseteq T'$ where T' is an MST in $G(V, E)$.