Problem Set 6

Ekkapot Charoenwanit
Efficient Algorithms

Problem 6.1. Fibonacci numbers

1) The pseudocode given in Algorithm 1 implements a function computing the Fibonacci numbers in a top-down manner with the help of memoization.

Algorithm 1 computes the Fibonacci numbers with memoization.

```
procedure FIB(n,F[1...n])
    if n ≤ 1 then
        return n
    else
        if F[n] > 0 then
            return F[n]
        F[n] = FIB(n − 1, F) + FIB(n − 2, F)
        return F[n]
```

What is the recursion depth and what is the space complexity of FIB(n)?

2) Optimize the bottom-up implementation given in Algorithm 2 such that it uses $\Theta(1)$ space.

Algorithm 2 computes the Fibonacci numbers bottom-up.

```
procedure FIB(n)
    F = NEW TABLE[0...n]
    F[0] = 0
    F[1] = 1
    for i = 2 → n do
        F[i] = F[i − 1] + F[i − 2]
    return F[n]
```

Problem 6.2. Matrix Chain Multiplication

1) Find an optimal parenthesization for the chain product of 5 matrices with dimensions 6×7, 7×8, 8×3, 3×10 and 10×6 in a bottom-up approach.

2) Show that the number of ways of parenthesization $C(n)$ is $\Omega(2^n)$, where n is the matrix chain length. You may use induction to show that $C(n) \geq c \cdot 2^n$ for some $c \in R^+$ and $n_0 \in Z^+$ for all $n \geq n_0$.

3) Calculate the exact number of distinct subproblems for a matrix chain of length n.

4) Does the maximum matrix chain problem also exhibit optimal substructure? If it is the case, prove your claim using a cut-and-paste argument.

Problem 6.3. Longest Common Subsequence

1) Give a memoized version of LCS-LENGTH that runs in $\mathcal{O}(mn)$ time.

2) What is the maximum recursion depth of LCS?