Final Practice Problems

1 Subset Sum

You are given a sequence of n numbers (positive or negative):

$$x_1, x_2, \ldots, x_n$$

Your job is to select a subset of these numbers of maximum total sum, subject to the constraint that you can't select two elements that are adjacent (that is, if you pick x_i then you cannot pick either x_{i-1} or x_{i+1}).
Explain how you can find, in time polynomial in n, the subset of maximum total sum.

2 Collecting Coins

You are given an n-by-n grid, where each square (i, j) contains $c(i, j)$ gold coins. Assume that $c(i, j) \geq 0$ for all squares. You must start in the upper-left corner and end in the lower-right corner, and at each step you can only travel one square down or right. When you visit any square, including your starting or ending square, you may collect all of the coins on that square. Give an algorithm to find the maximum number of coins you can collect if you follow the optimal path.

3 True/False

Decide whether these statements are **True** or **False**. You must briefly justify all your answers to receive full credit.

1. Any Dynamic Programming algorithm with n subproblems will run in $O(n)$ time.
   True **False**

   Explain:

2. Karatsuba's method is based on the use of continued fractions.
   True **False**

   Explain:

3. Newton's Method for computing $\sqrt{2}$ essentially squares the number of correct digits at each iteration.
   True **False**

   Explain:

4 Numerics

Suppose we are trying to compute $\sqrt[3]{9}$ (the cube root of 9).

Explain carefully how one iteration of Newton's Method works for this problem, starting with an initial guess of $x_0 = 2$. (Hint: the function to use is $f(x) = x^3 - 9$.) Be sure to derive carefully the value of x_1.