Introduction to Algorithms: 6.006
Massachusetts Institute of Technology
Professors Srini Devadas and Erik Demaine

May 9, 2008
Handout 13

Final Practice Problems

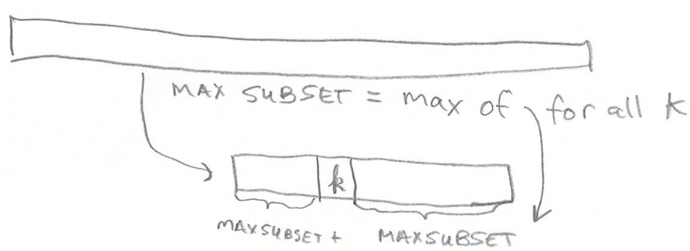
1 Subset Sum

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.

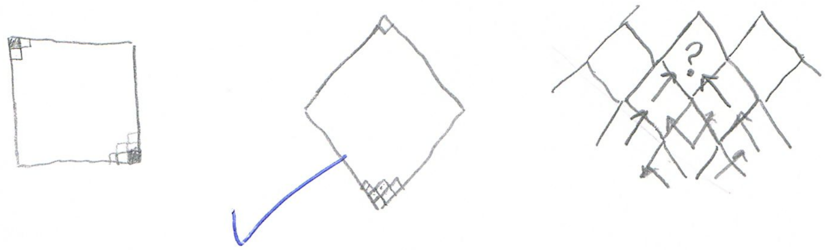
To find the maximum subset, we can leverage an optimal solution property to use dynamic programming. Here, we know that the max subset of some range of the sequence $[i,j]$ is the maximum of the pairs of max subsets found by deleting element k, $i \le k \le j$ and finding the max subset of both $[i, k-1]$ & $[k+1, j]$.



Since max subsets are contiguous regions there are $O(n^2)$ subproblems requiring linear time (to calculate max) to process for an $O(n^3)$ algorithm.
A dynamic programming approach would be to store the maxsubset of all $[i,j]$ regions in a table starting with all length-1 subsets then, length-2, so that the dependencies would be fulfilled in order.

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.



Using dynamic programming we can see that the optimal subproblem is the best path starting from (i,j) which is the max of the best path value in the bottom and right adjacent squares.

This problem can be solved in a bottom up fashion by having a table which stores, best-path-value (number of coins on maximal pa ending on (i,j)) and best path which is a list of moves correspondi to the best path value.

Then, starting from the bottom right corner fill in the first values $(n-1, n)$ & $(n, n-1)$ which have a maximal path value $= c(i,j) + c(n,n)$ and a path $((i,j),(n,n))$. From there expand each tile with its upper and left tiles (if they exist) using the formula:

$$\text{best-path-value}_{ij} = \max\left(c(i+1, j),\ c(i, j+1)\right)$$

$$\text{best-path} = (i^*, j^*) + \text{best-path}(i^*, j^*) \quad // \text{ where } i^*, j^* \text{ has the m}$$

This expansion occurs in a breadth-first search pattern so the algorithm never tries to compute a missing value.

The result is simply to return best-path-value for $(0,0)$ (or best-path if you wish to know which path results in that value)

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: If the subproblems take greater than constant time to compute, this will not hold.

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:

   $f(x) = x^2 - 2$

   This is because each iteration calculates the difference between x_i^2 and 2, the error term is repeatedly squared which is the same as doubling the power of 10, i.e.

   not squaring

   $(10^{-8})^2 = 10^{-16}$

   2x the precision, not 8x.

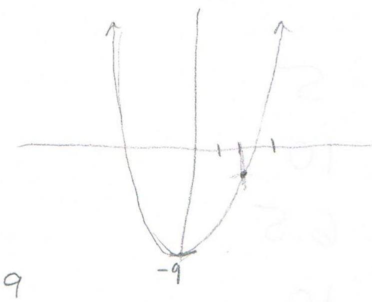
2.1
2.1
21
420
4.41

4.41
2.1
8.82
.441
9.261

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.



The cube root of 9 is where the function

$f(x) = x^3 - 9 = 0$

With an initial guess of 2, we see that

$f(x) = -1$

Newton's Method uses a linear approximation for $f(x)$ at $f(2)$ in order to calculate x_1

The linear approximation of $f(x)$ at $f(2)$ has the form

$y = mx + b$

$m = f'(2) = 3x^2 = 12$

$-1 = 12(2) + b$

$b = -25$

$\underline{y = 12x - 25}$

We now wish to calculate the x-intercept of this linear estimat

$0 = 12x - 25$

$25 = 12x$

$\boxed{x_1 = \frac{25}{12}}$

GRADE: (Assume each Q worth 10 pts)

1: 5

2: 10

3: 6.5

4: 10

31.5/40 = 79%

* Note, for Q1 the algorithm works but is considerably slower than the optimal algorithr (although still polynomial time) so I gave half marks. Note the grade of 66% if this solution were to be awarded zero points.