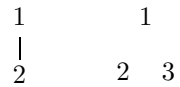
Homework 4 - Solutions
Fundamental Algorithms, Fall 2001
Professor Yap

November 28, 2001

1. (20 points) We will prove that the smallest AVL tree such that the deletion of a node in T will cause two rotation events is of height= 4 and size= 12 (this is the smallest AVL tree of height 4). Hence it suffices to show that deleting a node from an AVL tree of height 1, 2, 3 we do no more than one rotation.

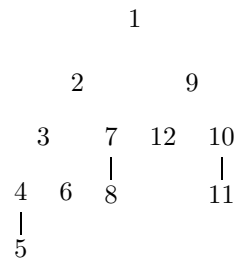
   For an AVL tree of height 1:

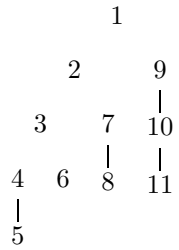
   a deletion of a node doesn't determine any imbalance.

   For an AVL tree of height 2 and 3 it is easy to see that after any deletion at most one rotation is necessary to rebalance the tree.

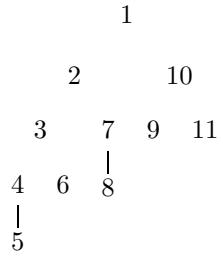
   So our next try is with the smallest AVL tree of height 4. One such tree is:

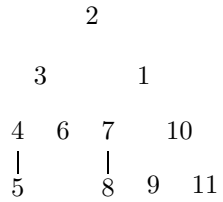
   If we delete 12 we get:



Now our tree is no more balanced at 9 so we do a left rotation at 9 and we get:



Again we get an imbalance, this time at 1. We do a right rotation at 1 and we get:



And this time the tree is balanced. Hence after 2 rotations the tree is balanced.

2. (15 points) We have 6 matrices with the following dimensions:

matrix	dimension
A_1	5×10
A_2	10×3
A_3	3×12
A_4	12×5
A_5	5×50
A_6	50×6

The tables m and s are computed as follows:

$$m[1,1] = 0 \quad m[2,2] = 0 \quad m[3,3] = 0$$
$$m[4,4] = 0 \quad m[5,5] = 0 \quad m[6,6] = 0$$

since no multiplication is necessary to compute A_i.

$m[1,2] = 5 \times 10 \times 3 = 150 \quad m[2,3] = 10 \times 3 \times 12 = 360 \quad m[3,4] = 3 \times 12 \times 5 = 180$
$m[4,5] = 12 \times 5 \times 50 = 3000 \quad m[5,6] = 5 \times 50 \times 6 = 1500$

Also: $s[1,2] = 1,\ s[2,3] = 2,\ s[3,4] = 3,\ s[4,5] = 4,\ s[5,6] = 5$

$$\begin{aligned} m[1,3] &= \min \left\{ \begin{array}{lcl} m[1,2] + p_0 \cdot p_2 \cdot p_3 &=& 150 + 180 = 330 \\ m[2,3] + p_0 \cdot p_1 \cdot p_3 &=& 360 + 600 = 960 \end{array} \right\} = 330 \\ &\Rightarrow\ s[1,3] = 2 \end{aligned}$$

$$\begin{aligned} m[2,4] &= \min \left\{ \begin{array}{lcl} m[2,3] + p_1 \cdot p_3 \cdot p_4 &=& 360 + 600 = 960 \\ m[3,4] + p_1 \cdot p_2 \cdot p_4 &=& 180 + 150 = 330 \end{array} \right\} = 330 \\ &\Rightarrow\ s[2,4] = 2 \end{aligned}$$

$$\begin{aligned} m[3,5] &= \min \left\{ \begin{array}{lcl} m[3,4] + p_2 \cdot p_4 \cdot p_5 &=& 180 + 750 = 930 \\ m[4,5] + p_2 \cdot p_3 \cdot p_5 &=& 3000 + 1800 = 4800 \end{array} \right\} = 930 \\ &\Rightarrow\ s[3,5] = 4 \end{aligned}$$

$$\begin{aligned} m[4,6] &= \min \left\{ \begin{array}{lcl} m[4,5] + p_3 \cdot p_5 \cdot p_6 &=& 3000 + 3600 = 6600 \\ m[5,6] + p_3 \cdot p_4 \cdot p_6 &=& 1500 + 360 = 1860 \end{array} \right\} = 1860 \\ &\Rightarrow\ s[4,6] = 4 \end{aligned}$$

$$\begin{aligned} m[1,4] &= \min \left\{ \begin{array}{lcl} m[1,3] + m[4,4] + p_0 \cdot p_3 \cdot p_4 &=& 330 + 300 = 630 \\ m[1,2] + m[3,4] + p_0 \cdot p_2 \cdot p_4 &=& 330 + 75 = 405 \\ m[1,1] + m[2,4] + p_0 \cdot p_1 \cdot p_4 &=& 330 + 250 = 580 \end{array} \right\} = 405 \\ &\Rightarrow\ s[1,4] = 2 \end{aligned}$$

$$\begin{aligned} m[2,5] &= \min \left\{ \begin{array}{lcl} m[2,4] + m[5,5] + p_1 \cdot p_4 \cdot p_5 &=& 330 + 2500 = 2830 \\ m[2,3] + m[4,5] + p_1 \cdot p_3 \cdot p_5 &=& 3360 + 6000 = 9360 \\ m[2,2] + m[3,5] + p_1 \cdot p_2 \cdot p_5 &=& 930 + 1500 = 2430 \end{array} \right\} = 2430 \\ &\Rightarrow\ s[2,5] = 2 \end{aligned}$$

$$\begin{aligned} m[3,6] &= \min \left\{ \begin{array}{lcl} m[3,5] + m[6,6] + p_2 \cdot p_5 \cdot p_6 &=& 930 + 900 = 1830 \\ m[3,4] + m[5,6] + p_2 \cdot p_4 \cdot p_6 &=& 1680 + 90 = 1770 \\ m[3,3] + m[4,6] + p_2 \cdot p_3 \cdot p_6 &=& 1880 + 216 = 2076 \end{array} \right\} = 1770 \\ &\Rightarrow\ s[3,6] = 4 \end{aligned}$$

$$\begin{aligned} m[1,5] &= \min \left\{ \begin{array}{lcl} m[1,4] + m[5,5] + p_0 \cdot p_4 \cdot p_5 &=& 405 + 1250 = 1655 \\ m[1,3] + m[4,5] + p_0 \cdot p_3 \cdot p_5 &=& 3330 + 3000 = 6330 \\ m[1,2] + m[3,5] + p_0 \cdot p_2 \cdot p_5 &=& 1080 + 750 = 1830 \\ m[1,1] + m[2,5] + p_0 \cdot p_1 \cdot p_5 &=& 2430 + 2500 = 4930 \end{array} \right\} = 1655 \\ &\Rightarrow\ s[1,5] = 4 \end{aligned}$$

$$m[2,6] = \min \left\{ \begin{array}{rcrcr} m[2,5]+m[6,6]+p_1\cdot p_5\cdot p_6 & = & 2430+3000 & = & 5280 \\ m[2,4]+m[5,6]+p_1\cdot p_4\cdot p_6 & = & 1830+300 & = & 2130 \\ m[2,3]+m[4,6]+p_1\cdot p_3\cdot p_6 & = & 2220+720 & = & 2940 \\ m[2,2]+m[3,6]+p_1\cdot p_2\cdot p_6 & = & 1770+180 & = & 1950 \end{array} \right\} = 1950$$

$$\Rightarrow \quad s[2,6] = 2$$

$$m[1,6] = \min \left\{ \begin{array}{rcrcr} m[1,5]+m[6,6]+p_0\cdot p_5\cdot p_6 & = & 1650+1500 & = & 3150 \\ m[1,4]+m[5,6]+p_0\cdot p_4\cdot p_6 & = & 1905+150 & = & 2055 \\ m[1,3]+m[4,6]+p_0\cdot p_3\cdot p_6 & = & 2190+360 & = & 2550 \\ m[1,2]+m[3,6]+p_0\cdot p_2\cdot p_6 & = & 1920+90 & = & 2010 \\ m[1,1]+m[2,6]+p_0\cdot p_1\cdot p_6 & = & 1860+300 & = & 2160 \end{array} \right\} = 2010$$

$$\Rightarrow \quad s[1,6] = 2$$

Using the values in the s-table we conclude that the optimal way to multiply the given matrices is

$(A_1 \times A_2)((A_3 \times A_4)(A_5 \times A_6))$.

3. (15 points) We have : $X = (0,1,0,1,1,0,1,1,0)$ and $Y = (1,0,0,1,0,1,0,1)$. We denote by X_i the prefix of length i of the string X for each $0 \leq i \leq 9$ and similarly Y_j for each $0 \leq j \leq 8$.

Trivially , $c[0,j] = c[i,0] = 0$ for every $0 \leq i \leq 9$ and $0 \leq j \leq 8$.

Using the recurrence:

$$c[i,j] = \left\{ \begin{array}{rl} c[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } x_i = y_j. \\ \max(c[i-1,j], c[i,j-1]) & \text{if } i,j>0 \text{ and } x_i \neq y_j. \end{array} \right.$$

we fill the following table in row-major order. (the first row from left to right, then the second row and so on).

	j	0	1	2	3	4	5	6	7	8
i		y_j	1	0	0	1	0	1	0	1
0	x_i	0	0	0	0	0	0	0	0	0
1	0	0	$\uparrow$ 0	$\nwarrow$ 1	$\nwarrow$ 1	$\leftarrow$ 1	$\nwarrow$ 1	$\leftarrow$ 1	$\nwarrow$ 1	$\leftarrow$ 1
2	1	0	$\nwarrow$ 1	$\uparrow$ 1	$\uparrow$ 1	$\nwarrow$ 2	$\leftarrow$ 2	$\nwarrow$ 2	$\leftarrow$ 2	$\nwarrow$ 2
3	0	0	$\uparrow$ 1	$\nwarrow$ 2	$\nwarrow$ 2	$\uparrow$ 2	$\nwarrow$ 3	$\leftarrow$ 3	$\nwarrow$ 3	$\leftarrow$ 3
4	1	0	$\nwarrow$ 1	$\uparrow$ 2	$\uparrow$ 2	$\nwarrow$ 3	$\uparrow$ 3	$\nwarrow$ 4	$\leftarrow$ 4	$\nwarrow$ 4
5	1	0	$\nwarrow$ 1	$\uparrow$ 2	$\uparrow$ 2	$\nwarrow$ 3	$\uparrow$ 3	$\nwarrow$ 4	$\uparrow$ 4	$\nwarrow$ 5
6	0	0	$\uparrow$ 1	$\nwarrow$ 2	$\nwarrow$ 3	$\uparrow$ 3	$\nwarrow$ 4	$\uparrow$ 4	$\nwarrow$ 5	$\uparrow$ 5
7	1	0	$\nwarrow$ 1	$\uparrow$ 2	$\uparrow$ 3	$\nwarrow$ 4	$\uparrow$ 4	$\nwarrow$ 5	$\uparrow$ 5	$\nwarrow$ 6
8	1	0	$\nwarrow$ 1	$\uparrow$ 2	$\uparrow$ 3	$\nwarrow$ 4	$\uparrow$ 4	$\nwarrow$ 5	$\uparrow$ 5	$\nwarrow$ 6
9	0	0	$\uparrow$ 1	$\nwarrow$ 2	$\nwarrow$ 3	$\uparrow$ 4	$\nwarrow$ 5	$\uparrow$ 5	$\nwarrow$ 6	$\uparrow$ 6

To build the LCS(X,Y) one has to follow the arrows from the lower right-hand corner. In our case we get :

LCS(X,Y)=$< 0,1,0,1,0,1 >$.

4. (25 points) Let X be the input sequence of numbers and X_i be the prefix of X of length i. Suppose for each list X_i we keep a list $L_i = (a_1 \leq a_2 \leq \ldots \leq a_{j(i)})$ of numbers such that a_k for each $1 \leq k \leq j(i)$ has the properties:

   (a) there exists a monotone subsequence of length k in X_i that ends in a_k.

   (b) if there are more monotone subsequences of length k in X_i then a_k is the smallest endpoint of all such subsequences.

   Note: a_1 is the smallest element in the X_i. We start to compute the values $L_1, \ldots, L_n$. Finally the length of L_n is the length of the longest monotone subsequence in X. We have: $L_1 = x(1)$. Let's try to find a way to derive L_{i+1} from L_i. We have: $X_{i+1} = X_i \cup x(i+1)$. Since the elements in L_i are in increasing order there exists a unique position l in L_i such that $a_l \leq x(i+1)$ and l is maximum with this property (if $x(i+1) < a_1$ we take $l = 0$).

   If $l < j(i)$ we replace a_{l+1} by $x(i+1)$.The idea behind this is that in X_i we have a monotone subsequence of length l ending in a_l. Since $a_l \leq x(i+1)$ we get a monotone subsequence of length $l+1$ in X_{i+1} with endpoint $x(i+1)$. On the other hand $x(i+1) < a_{l+1}$ (which is the smallest endpoint of a monotone subsequence of length l in X_i) , hence we have to update a_{l+1} to the value $x(i+1)$. In order to be able to rebuild the longest monotone subsequence (not only its length, but the path also) we keep a pointer from $x(i+1)$ to a_l (if $l = 0$, then this pointer is nil).

   If $l = j(i)$ then we add a $x(i+1)$ to the end of the list L_i and we get L_{i+1}. We keep also a pointer from $x(i+1)$ to $a_{j(i)}$.

   We claim that the complexity of the algorithm is $O(n^2)$. Indeed, for each element $x(i)$ in the list X we compare $x(i)$ to at most $j(i) \leq i \leq n$ elements, hence for all n elements in X we have $O(n^2)$ comparisons. And from this results our claim.

Optional Exercises

1. Let X and Y be the two input strings. Let X_i and Y_j denote their respective prefixes of length i and j. Let $D(X_i, Y_j)$ be the edit distance between X_i and Y_j , but we don't allow kill at this moment. We look for a recursive definition for $D(X_i, Y_j)$. We have: $D(X_i, Y_0) = i \cdot \text{cost(delete)}$ where Y_0 is the empty string and $0 \leq i \leq m$. $D(X_0, Y_j) = j \cdot \text{cost(insert)}$ where X_0 is the empty string and $0 \leq j \leq n$. For $0 < i \leq m$ and $0 < j \leq n$ we have the recursive formula:

$$D(X_i, Y_j) = \min \begin{cases} D(X_{i-1}, Y_{j-1} + cost(copy) & \text{if } x(i) = y(j) \\ D(X_{i-1}, Y_{j-1} + cost(replace) & \\ D(X_{i-1}, Y_j) + cost(delete) & \\ D(X_i, Y_{j-1}) + cost(insert) & \\ D(X_{i-2}, Yj-2) + cost(twiddle) & \text{if } x(i) = y(j-1) \\ & \text{and } x(i-1) = y(j) \end{cases}$$

Now to compute $D(X, Y)$ with all operations (including kill) we have the formula:

$$EditDistance(X, Y) = \min(D(X_m, Y_n), \min_i(D(X_i, Y_n) + cost(kill)))$$

. We notice that we have to fill all the entries $D(i, j)$ for all $0 \leq i \leq m$ and $0 \leq j \leq n$. So the space requirement is $O(mn)$. In order to fill an entry $D(i, j)$ we need $O(1)$ time, hence the running time is $O(mn)$.