Linear Recurrences

Linear Recurrences

- A recurrence relation is a function that defines a sequence recursively.
  - $F_n = F_{n-1} + F_{n-2}$ (Fibonacci recurrence)
  - $x_n = r.x_{n-1}(1 - x_{n-1})$ (Logistic map)
- A linear recurrence simply consists of a linear combination of some number of preceding terms.
  - $T_n = \boldsymbol{a} \cdot \boldsymbol{v}$ where $\boldsymbol{a}$ is a vector of coefficients and $\boldsymbol{v}$ is a vector such that $\boldsymbol{v_i} = T_{n-k}$ for some $k \in \mathbb{N}^*$

Linear Recurrences

- How do we solve linear recurrences efficiently?
- For example take the Fibonacci numbers:
  - $F_n = F_{n-1} + F_{n-2}$
  - $F_1 = F_2 = 1$
- How do we compute $F_{1000\,000\,000}$?
  - DP? Too slow.
  - We can do much better using matrices.

Matrices

- What is a matrix?
  - A matrix is simply a block of numbers that can be added, subtracted and multiplied as if it were a number in its own right.
  - They come in different sizes; an $m \times n$ matrix has m rows and n columns.
  - If $m = n$, it is square matrix.
  - A matrix with only 1 row is called a row vector and one with only 1 column is called a column vector.

Matrices

$$\begin{pmatrix} 1 & 7 \\ -3 & 1/2 \end{pmatrix}$$

2x2 square matrix

$$\begin{pmatrix} 6 & -79 \\ \text{٣}/4 & 6 \\ -3 & 37 \end{pmatrix}$$

3x2 matrix

$$\begin{pmatrix} 5 & -\text{٤} & \text{٨} & 17 \end{pmatrix}$$

1x4 row vector

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 11 & 0 \\ 0 & 0 & 0 & 16 \end{pmatrix}$$

4x4 square matrix

$$\begin{pmatrix} -1 & 75 & 15 \\ 0 & 6 & -57 \end{pmatrix}$$

2x3 matrix

Matrices

- The matrix analogue of zero is the zero matrix and the analogue of one is the identity matrix.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

4x3 zero matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

5x5 identity matrix

Matrices

- When adding or subtracting matrices you simply add or subtract their individual terms.
- Matrix multiplication however is a bit trickier and it forms the basis of this talk.

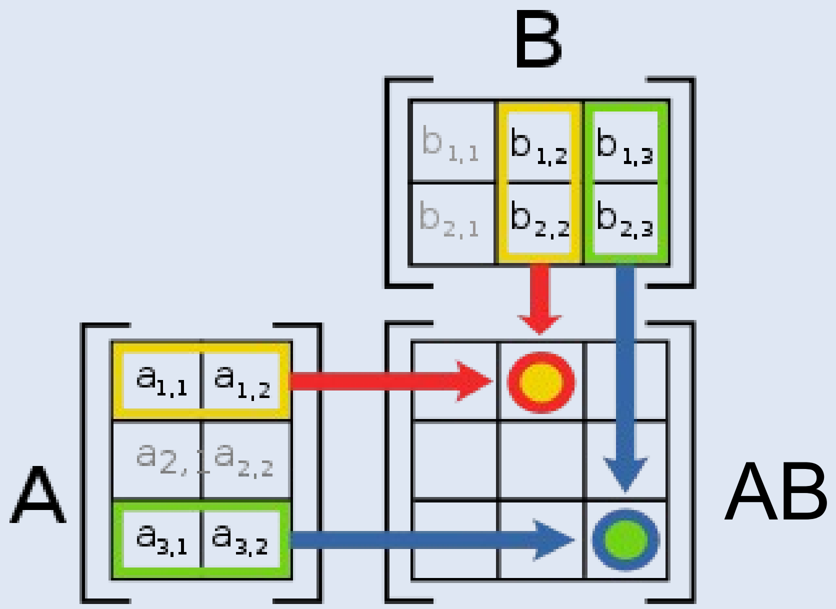
Matrix Multiplication

- Matrices may only be multiplied if the number of columns in the first is equal to the number of rows in the second. The result has the same number rows as the first and the same number of columns as the second.

$$(m \times \underbrace{n)\cdot(n}_{} \times p) = (m \times p)$$

Matrix Multiplication

- If two matrices are multiplied each element in the resulting matrix is the dot product of the corresponding row in the first matrix and the corresponding column in the second matrix.



Matrix Multiplication

- Matrix multiplication is not commutative, but it is associative and it distributes over matrix addition.

$$AB \neq BA$$

$$(AB)C = A(BC)$$

$$A(B+C) = AB + AC$$

$$(A+B)C = AC + BC$$

Matrix Multiplication

$$\begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \text{ is undefined}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}\begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix} = \begin{pmatrix} 1\times 7+2\times 8+3\times 9 \\ \text{٤}\times \text{٧}+5\times 8+\text{٦}\times 9 \end{pmatrix} = \begin{pmatrix} 50 \\ \text{١٢٢} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1\times 3+2\times 1 & 1\times 2+2\times 0 \\ 3\times 3+4\times 1 & 3\times 2+4\times 0 \end{pmatrix} = \begin{pmatrix} 5 & 2 \\ 13 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 3\times 1+2\times 3 & 3\times 2+2\times 4 \\ 1\times 1+0\times 3 & 1\times 2+0\times 4 \end{pmatrix} = \begin{pmatrix} 9 & 14 \\ 1 & 2 \end{pmatrix}$$

Matrix Multiplication

```
// O(n^3) matrix multiplication algorithm
Matrix multiply( Matrix a, Matrix b ) {

   assert( a.columns == b.rows );

   // c is initially filled with zeros
   Matrix c( a.rows, b.columns );

   for( int k = 0; k < a.columns; k ++ ) {

      for( int i = 0; i < a.rows; i ++ ) {

         for( int j = 0; j < b.columns; j ++ ) {

            c[i][j] += a[i][k]*b[k][j];
         }
      }
   }

   return c;
}
```

Recurrences in terms of Matrices

- Write down the equation

$$F_n = F_{n-1} + F_{n-2}$$

- What information is used?

$$\begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}$$

- What do we want and what will be used next iteration?

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix}$$

Recurrences in terms of Matrices

- What matrix **A** do you need to multiply the first column vector by to get the second one?

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = A \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}$$

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix} \qquad \begin{aligned} F_n &= a \cdot F_{n-1} + b \cdot F_{n-2} \\ F_{n-1} &= c \cdot F_{n-1} + d \cdot F_{n-2} \end{aligned}$$

Recurrences in terms of Matrices

- What matrix **A** do you need to multiply the first column vector by to get the second one?

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}$$

- Given

$$F_n = F_{n-1} + F_{n-2}$$

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}$$

$$F_n = 1 \cdot F_{n-1} + 1 \cdot F_{n-2}$$
$$F_{n-1} = 1 \cdot F_{n-1} + 0 \cdot F_{n-2}$$

Recurrences in terms of Matrices

Because matrix multiplication is associative, if we want the 6th Fibonacci number we can say

$$\begin{pmatrix} F_6 \\ F_5 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} F_2 \\ F_1 \end{pmatrix}$$

- Now we can simply power the matrix!

$$\begin{pmatrix} F_6 \\ F_5 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^4 \begin{pmatrix} F_2 \\ F_1 \end{pmatrix}$$

$$\begin{pmatrix} F_{٦} \\ F_{٥} \end{pmatrix} = \begin{pmatrix} ١ & 1 \\ ١ & 0 \end{pmatrix}^4 \begin{pmatrix} ١ \\ 1 \end{pmatrix}$$

Recurrences in terms of Matrices

- This can be made very fast with the fast exponentiation algorithm from Michel's lecture modified to work for matrices.
- But you don't even have to write the code, just implement a matrix class with an overloaded multiplication operator and call __gnu_cxx::power from <ext/numeric>.
- This runs in $\Theta(m^3 \log n)$ where *n* is the power and *m* is the size of the matrix. Here *m* is a constant so it runs in $\Theta(\log n)$!

A more complicated example

- Find the nth term of the following series (given c):

$$x_n=\sum_{i=1}^{n}(i\cdot c^{i-1}) = 1+2c+3c^2+...+nc^{n-1}$$

- Phrase it as a recurrence:

$$x_i = x_{i-1}+i\,c^{i-1}$$

- Used info

$$\begin{pmatrix} x_{i-1} \\ ic^{i-1} \end{pmatrix}$$

A more complicated example

- Equation

$$\begin{pmatrix} x_i \\ (i+۱)c^i \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} x_{i-1} \\ ic^{i-1} \end{pmatrix}$$

$$\begin{pmatrix} x_i \\ (i+1)c^i \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ ? & ? \end{pmatrix} \begin{pmatrix} x_{i-1} \\ ic^{i-1} \end{pmatrix} \qquad \begin{aligned} x_i &= ۱ \cdot x_{i-1} + 1 \cdot i\,c^{i-1} \\ (i+1)c^i &= ? \cdot x_{i-1} + ? \cdot i\,c^{i-1} \end{aligned}$$

- But ic^{i-1} cannot be maintained, so an extra variable must be added.

$$\begin{pmatrix} x_i \\ (i+1)c^i \\ c^i \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} x_{i-1} \\ ic^{i-1} \\ c^{i-1} \end{pmatrix} \quad \longleftarrow$$

Added c^{i-1}

A more complicated example

- Now solving is easy

$$\begin{pmatrix} x_i \\ (i+١)c^i \\ c^i \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & c & c \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} x_{i-1} \\ ic^{i-1} \\ c^{i-1} \end{pmatrix}$$

$$x_i = 1 \cdot x_{i-1} + ١ \cdot i\,c^{i-1} + 0 \cdot c^{i-1}$$
$$(i+1)c^i = 0 \cdot x_{i-1} + c \cdot i\,c^{i-1} + c \cdot c^{i-1}$$
$$c^i = 0 \cdot x_{i-1} + 0 \cdot i\,c^{i-1} + c \cdot c^{i-1}$$

$$\begin{pmatrix} x_n \\ (n+1)c^n \\ c^n \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & c & c \\ 0 & 0 & c \end{pmatrix}^n \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

$$x_0 = 0$$
$$(0+1)c^0 = 1$$
$$c^0 = 1$$

Notable Mentions

- If **A** is the adjacency matrix of a graph. Then the element (i,j) of $\mathbf{A}^n$ is the number of distinct paths of length *n* from i to j.
- The number of paths up to length *n* can be found by summing the matrices from $\mathbf{A}^0$ to $\mathbf{A}^n$. The matrix geometric sum formula can be use to compute this efficiently.

$$S_n = (A^{n+1} - I)(A - I)^{-1} \quad \text{where } I \text{ is the identity matrix}$$

Example – Google code jam

- The question is simple: find the last 3 digits before the decimal point of $(3+\sqrt{5})^n$. For $n \in \mathbb{N}$
  - Google code jam, Round 1A, Question C

- For example, $(3+\sqrt{5})^5 = 3935.73982\ldots$, so you output 935.

Example – Google code jam

- The answer is less simple.
- Doing floating point operations loses accuracy so you get the wrong answer.
- We can only use integers in our algorithm.

Example – Google code jam

- If we represent $(3+\sqrt{5})^n$ as $a_n + b_n\sqrt{5}$; $a, b \in \mathbb{Z}$ (which is always possible), we can get the following recurrence:

$$a_n = 3a_{n-1} + 5b_{n-1}$$
$$b_n = a_{n-1} + 3b_{n-1}$$

- Written as a matrix

$$\begin{pmatrix} a_n \\ b_n \end{pmatrix} = \begin{pmatrix} 3 & 5 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} a_{n-1} \\ b_{n-1} \end{pmatrix}$$

Example – Google code jam

- If you need to turn a surd into an integer you usually perform some operation with its conjugate.
- The conjugate surd of $3+\sqrt{5}$ is $3-\sqrt{5}$.

Example – Google code jam

- The key observation is as follows.
  - let $x_n = (3+\sqrt{5})^n + (3-\sqrt{5})^n$
  - Because $\text{٣}-\sqrt{5}$ is less than 1, for positive *n*, $(3-\sqrt{\text{٥}})^n$ will also be less than 1.
  - As we will show, x_n will always be an integer.
  - Therefore $x_n = \lceil (3+\sqrt{5})^n \rceil$, which is one more than our desired answer.

Example – Google code jam

$$x_n = \sum_{i=0}^{n} \binom{n}{i} \cdot 3^{n-i} \cdot \sqrt{5}^{i} + \sum_{i=0}^{n} \binom{n}{i} \cdot 3^{n-i} \cdot \sqrt{5}^{i} \cdot (-1)^{i}$$

Binomial theorem

$$x_n = \sum_{i=0}^{n} \binom{n}{i} \cdot 3^{n-i} \left(\sqrt{5}^{i} + \sqrt{5}^{i} \cdot (-1)^{i} \right)$$

Factorise

$$x_n = 2 \cdot \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{2i} \cdot 3^{n-2i} \cdot 5^{i} + \sum_{i=1}^{\lceil n/2 \rceil} \binom{n}{2i-1} \cdot 3^{n+1-2i} \cdot \left(\sqrt{5^{2i-1}} - \sqrt{5^{2i-1}} \right)$$

Split sum into even and odd parts

$$x_n = 2 \cdot \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{2i} \cdot 3^{n-2i} \cdot 5^{i} + 0$$

Simplify

$$x_n = 2 a_n$$

From original binomial expansion noting that odd terms have a root 5 factor and even terms don't

Example – Google code jam

- Therefore we compute the nth power of the matrix, double the top left-hand item and subtract one. (All these operation are computed mod 1000).

$$2\left[\begin{pmatrix}3 & 5\\ 1 & 3\end{pmatrix}^{n}\right]_{1,1}-1 \pmod{\text{١٠٠٠}}$$

Example – SACO

- How many ways are there to tile a 1xN corridor with 1x2 and 1x3 tiles?
(Give the answer mod 1 000 000 009)
  - SACO 2nd round 2008, Question 4 (modified)

Example – SACO

- You all solved it with $N <= 1\,000\,000$
- But, just like the Fibonacci numbers, this can be solved with matrices for $N <= 10^{1000\,000}$
- This same approach can be used to solve the problems *jelly* and *tile* from the 2009 on-line contest in logarithmic time.

Questions

- Any Questions?