INF01056
Aula 17/18
Matemática

Prof. João Comba

Baseado no Livro Programming Challenges

Prime Numbers

- Prime number: Integer p > 1 such that
  - only divisible by 1
  - p = a * b
  - 2, 3, 5, 7, 11, 13, 17, 19, 23, 27, ...
- Fundamental Theorem of Arithmetic
  - Every integer can be expressed in only one way as the product of primes
  - 105 = 3 x 5 x 7 32 = 2 x 2 x 2 x 2 x 2

Finding Primes

- Only even prime is 2
- test only odd numbers up to $\sqrt{n}$.

Finding Primes

- Only even prime is 2
- test only odd numbers up to $\sqrt{n}$
- Proof (By contradiction):
  - suppose n is composite with a factor p larger than $\sqrt{n}$
  - n / p must divide n, and also be > $\sqrt{n}$
  - but then $p * n / p > n$ CONTRADICTION

Finding Prime Factors

```
prime_factorization(long x) {
   long i;      /* counter */
   long c;      /* remaining product to factor */

   c = x;
   while ((c % 2) == 0) {
      printf("%ld\n",2);
      c = c / 2;
   }

   i = 3;
   while (i <= (sqrt(c)+1)) {
      if ((c % i) == 0) {
         printf("%ld\n",i);
          c = c / i;
      }
      else
         i = i + 2;
          }

   if (c > 1) printf("%ld\n",c);
}
```

Counting Primes

- How many primes are there ?

Counting Primes

- How many primes are there ?
  - Suppose finite number of primes $p_1, p_2, \ldots, p_n$
  - Let $m = 1 + \prod_{i=1}^{n} p_i$, be a composite number
  - Which prime divide it ?
    - m leaves a reminder of 1 for all primes
    - Therefore, m is also prime CONTRADICTION

The Number of Prime Numbers is Infinite

Counting Primes

Sieve of Eratosthenes

To find all the prime numbers less than or equal to 30, proceed as follows.

First generate a list of integers from 2 to 30:

```
 2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
```

First number in the list is 2; cross out every 2nd number in the list after it (by counting up in increments of 2), i.e. all the multiples of 2:

```
 2  3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25 ~~26~~ 27 ~~28~~ 29 ~~30~~
```

Next number in the list after 2 is 3; cross out every 3rd number in the list after it (by counting up in increments of 3), i.e. all the multiples of 3:

```
 2  3 ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24~~ 25 ~~26 27 28~~ 29 ~~30~~
```

Next number not yet crossed out in the list after 3 is 5; cross out every 5th number in the list after it (by counting up in increments of 5), i.e. all the multiples of 5:

```
 2  3 ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24 25 26 27 28~~ 29 ~~30~~
```

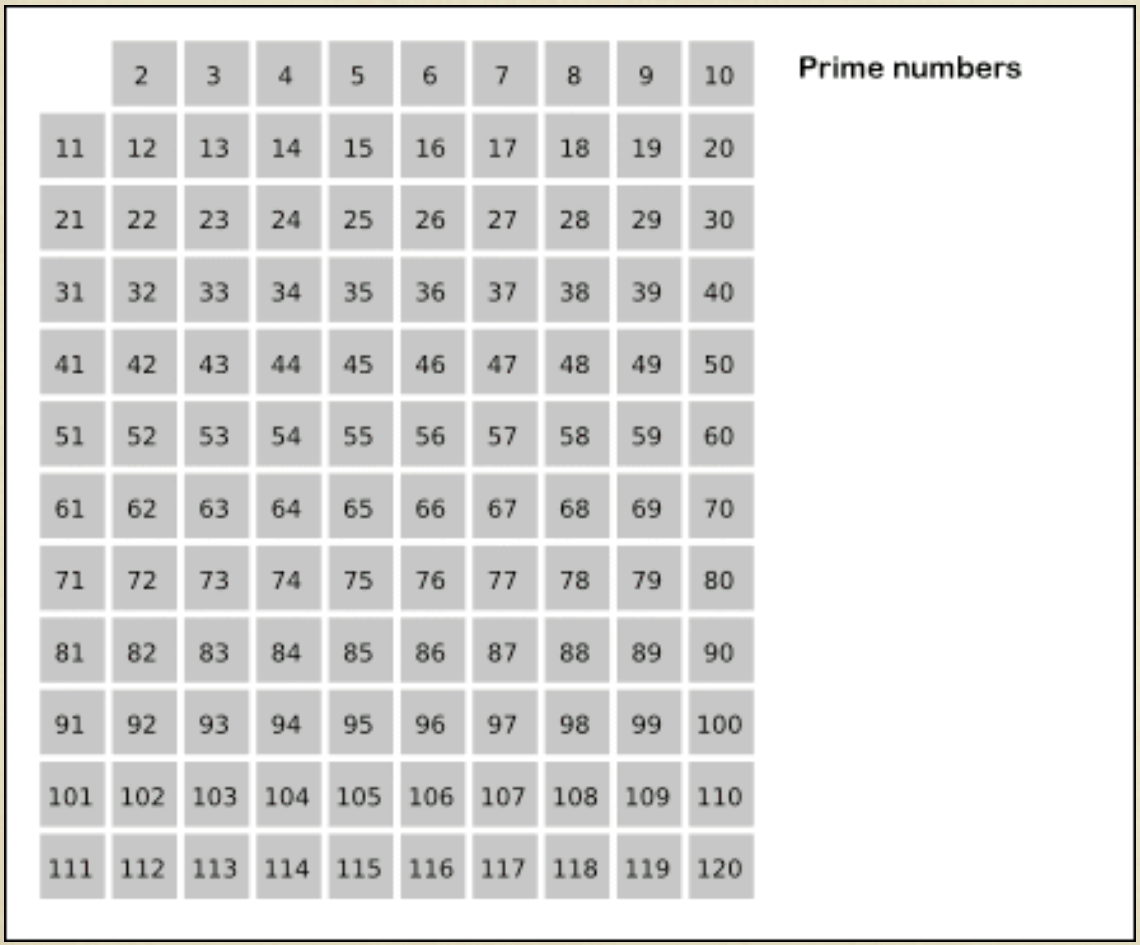
Next number not yet crossed out in the list after 5 is 7; the next step would be to cross out every 7th number in the list after it, but they are all already crossed out at this point, as these numbers (14, 21, 28) are also multiples of smaller primes because 7*7 is greater than 30. The numbers left not crossed out in the list at this point are all the prime numbers below 30:

```
 2  3     5     7           11    13          17    19          23                29
```

http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

Counting Primes

Sieve of Eratosthenes



http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

Divisibility

- Integer divisibility

$$b \text{ divides } a \qquad a = bk \text{ for some integer } k$$

- How to find all divisors of a given integer ?
  - find prime factorization
  - form all subsets of prime factors:
    - 12: 2 * 2 * 3 (1,2,3,4,6,12)

Greatest Common Divisor (GCD or MDC)

- largest divisor shared by a pair of integers
- GCD(x,y) = 1, x and y are relatively prime
- Euclid's Algorithm:
  - If B divides A, A = BK, therefore:
    - GCD(bk,b) = b or GCD(a,b) = b
  - If a = bt + r for integers t and r
    - GCD(A, B) = GCD(B, R)

Greatest Common Divisor (GCD or MDC)

- Largest divisor shared by a pair of integers
- GCD(x,y) = 1, x and y are relatively prime
- Euclid's Algorithm:
  - If B divides A, A = BK, therefore:
    - GCD(bk,b) = b or GCD(a,b) = b
  - If a = bt + r for integers t and r
    - GCD(A, B) = GCD(B, R)
    - PROOF: GCD(A, B) = GCD(BT+R, B)
      - But BT is divisible by any divisor of B

Greatest Common Divisor (GCD or MDC)

Let $a = 34398$ and $b = 2132$.

$$\begin{aligned} gcd(34398, 2132) &= gcd(34398 \bmod 2132, 2132) = gcd(2132, 286) \\ gcd(2132, 286) &= gcd(2132 \bmod 286, 286) = gcd(286, 130) \\ gcd(286, 130) &= gcd(286 \bmod 130, 130) = gcd(130, 26) \\ gcd(130, 26) &= gcd(130 \bmod 26, 26) = gcd(26, 0) \end{aligned}$$

Therefore, $gcd(34398, 2132) = 26$.

Greatest Common Divisor (GCD or MDC)

$$a \cdot x + b \cdot y = gcd(a, b)$$

which will prove quite useful in solving linear congruences. We know that $gcd(a, b) = gcd(b, a')$, where $a' = a - b\lfloor a/b \rfloor$. Further, assume we know integers x' and y' such that

$$b \cdot x' + a' \cdot y' = gcd(a, b)$$

by recursion. Substituting our formula for a' into the above expression gives us

$$b \cdot x' + (a - b\lfloor a/b \rfloor) \cdot y' = gcd(a, b)$$

$$34398 \times 15 + 2132 \times -242 = 26$$

Greatest Common Divisor (GCD or MDC)

```
long gcd1(long p, long q) {
  if (q > p) return(gcd1(q,p));
  if (q == 0) return(p);
  return( gcd1(q, p % q) );
}
```

Greatest Common Divisor (GCD or MDC)

```
long gcd(long p, long q, long *x, long *y) {
   long x1,y1;        /* previous coefficients */
   long g;          /* value of gcd(p,q) */

   if (q > p) return(gcd(q,p,y,x));

   if (q == 0) {
      *x = 1;
      *y = 0;
      return(p);
   }

   g = gcd(q, p%q, &x1, &y1);

   *x = y1;
   *y = (x1 - floor(p/q)*y1);

   return(g);
}
```

Least Common Multiple (LCM) - MMC

When is the next year (after 2000) that the presidencial election (which happens every 4 years) will coincide with census (which happens every 10 years) ? LCM(4,10)=20

Least Common Multiple (lcm or mmc)

$$lcm(x, y) \geq \max(x, y)$$

$$lcm(x, y) \leq xy$$

$$lcm(x, y) = xy/gcd(x, y)$$

Addition

- *Addition* — What is $(x + y) \bmod n$? We can simplify this to

$$((x \bmod n) + (y \bmod n)) \bmod n$$

to avoid adding big numbers. How much small change will I have if given \$123.45 by my mother and \$94.67 by my father?

$$(12,345 \bmod 100) + (9,467 \bmod 100) = (45 + 67) \bmod 100 = 12 \bmod 100$$

Subtraction

- *Subtraction* — Subtraction is just addition with negative values. How much small change will I have after spending \$52.53?

$$(12 \bmod 100) - (53 \bmod 100) = -41 \bmod 100 = 59 \bmod 100$$

Notice how we can convert a negative number mod n to a positive number by adding a multiple of n to it. Further, this answer makes sense in this change example. It is usually best to keep the residue between 0 and $n - 1$ to ensure we are working with the smallest-magnitude numbers possible.

Multiplication

- *Multiplication* — Since multiplication is just repeated addition,

$$xy \bmod n = (x \bmod n)(y \bmod n) \bmod n$$

How much change will you have if you earn \$17.28 per hour for 2,143 hours?

$$(1,728 \times 2,143) \bmod 100 = (28 \bmod 100) \times (43 \bmod 100) = 4 \bmod 100$$

Further, since exponentiation is just repeated multiplication,

$$x^y \bmod n = (x \bmod n)^y \bmod n$$

Since exponentiation is the quickest way to produce really large integers, this is where modular arithmetic really proves its worth.

Finding the Last Digit

- *Finding the Last Digit* — What is the last digit of 2^{100}? Sure we can use infinite precision arithmetic and look at the last digit, but why? We can do this computation by hand. What we really want to know is what $2^{100} \bmod 10$ is. By doing repeated squaring, and taking the remainder $\bmod 10$ at each step we make progress very quickly:

$$\begin{aligned}
2^3 \bmod 10 &= 8 \\
2^6 \bmod 10 &= 8 \times 8 \bmod 10 \to 4 \\
2^{12} \bmod 10 &= 4 \times 4 \bmod 10 \to 6 \\
2^{24} \bmod 10 &= 6 \times 6 \bmod 10 \to 6 \\
2^{48} \bmod 10 &= 6 \times 6 \bmod 10 \to 6 \\
2^{96} \bmod 10 &= 6 \times 6 \bmod 10 \to 6 \\
2^{100} \bmod 10 &= 2^{96} \times 2^3 \times 2^1 \bmod 10 \to 6
\end{aligned}$$

Light, More Light

The Problem

There is man named "mabu" for switching on-off light in our University. He switches on-off the lights in a corridor. Every bulb has its own toggle switch. That is, if it is pressed then the bulb turns on. Another press will turn it off. To save power consumption (or may be he is mad or something else) he does a peculiar thing. If in a corridor there is `n' bulbs, he walks along the corridor back and forth `n' times and in i'th walk he toggles only the switches whose serial is divisable by i. He does not press any switch when coming back to his initial position. A i'th walk is defined as going down the corridor (while doing the peculiar thing) and coming back again.

Now you have to determine what is the final condition of the last bulb. Is it on or off?

The Input

The input will be an integer indicating the n'th bulb in a corridor. Which is less then or equals 2^32-1. A zero indicates the end of input. You should not process this input.

The Output

Output "`yes`" if the light is on otherwise "`no`" , in a single line.

Sample Input

```
3
6241
8191
0
```

Sample Output

```
no
yes
no
```

Light, More Light

```cpp
#include <iostream>
#include <cmath>

using namespace std;

int main() {
  unsigned int n;
  double raiz;

  cin >> n;

  while (n != 0) {
        raiz = sqrt(n);
        if (floor(raiz) == ceil(raiz))
            cout << "yes" << endl;
        else
            cout << "no" << endl;
        cin.ignore();
        cin >> n;
  }
  return 0;
}
```

Bases Numéricas

- *Binary* — Base-2 numbers are made up of the digits 0 and 1. They provide the integer representation used within computers, because these digits map naturally to on/off or high/low states.
- *Octal* — Base-8 numbers are useful as a shorthand to make it easier to read binary numbers, since the bits can be read off from the right in groups of three. Thus $10111001_2 = 371_8 = 249_{10}$. They also play a role in the only base-conversion joke ever written. Why do programmers think Christmas is Halloween? Because 31 Oct = 25 Dec!
- *Decimal* — We use base-10 numbers because we learned to count on our ten fingers. The ancient Mayan people used a base-20 number system, presumably because they counted on both fingers and toes.
- *Hexadecimal* — Base-16 numbers are an even easier shorthand to represent binary numbers, once you get over the fact that the digits representing 10 through 15 are "A" to ''F."
- *Alphanumeric* — Occasionally, one sees even higher numerical bases. Base-36 numbers are the highest you can represent using the 10 numerical digits with the 26 letters of the alphabet. Any integer can be represented in base-X provided you can display X different symbols.

Conversões

Convert x_a to y_b

- *Left to Right* — Here, we find the most-significant digit of y first. It is the integer d_l such that

$$(d_l + 1)b^k > x \geq d_l b^k$$

where $1 \leq d_l \leq b - 1$. In principle, this can be found by trial and error, although you must to be able to compare the magnitude of numbers in different bases. This is analogous to the long-division algorithm described above.

- *Right to Left* — Here, we find the least-significant digit of y first. This is the remainder of x divided by b. Remainders are exactly what is computed when doing modular arithmetic in Section 7.3. The cute thing is that we can compute the remainder of x on a digit-by-digit basis, making it easy to work with large integers.

Números

Rational Numbers

Exact rational numbers x/y are best represented by pairs of integers x, y, where x is the *numerator* and y is the *denominator* of the fraction.

The basic arithmetic operations on rationals $c = x_1/y_1$ and $d = x_2/y_2$ are easy to program:

- *Addition* — We must find a common denominator before adding fractions, so

$$c + d = \frac{x_1 y_2 + x_2 y_1}{y_1 y_2}$$

- *Subtraction* — Same as addition, since $c - d = c + -1 \times d$, so

$$c - d = \frac{x_1 y_2 - x_2 y_1}{y_1 y_2}$$

- *Multiplication* — Since multiplication is repeated addition, it is easily shown that

$$c \times d = \frac{x_1 x_2}{y_1 y_2}$$

- *Division* — To divide fractions you multiply by the *reciprocal* of the denominator, so

$$c/d = \frac{x_1}{y_1} \times \frac{y_2}{x_2} = \frac{x_1 y_2}{x_2 y_1}$$

Álgebra

Manipulating Polynomial

- *Evaluation* — Computing $P(x)$ for some given x can easily be done by brute force, namely, computing each term $c_i x^n$ independently and adding them together. The trouble is that this will cost $O(n^2)$ multiplications where $O(n)$ suffice. The secret is to note that $x^i = x^{i-1}x$, so if we compute the terms from smallest degree to highest degree we can keep track of the current power of x, and get away with two multiplications per term ($x^{i-1} \times x$, and then $c_i \times x^i$).

  Alternately, one can employ *Horner's rule*, an even slicker way to do the same job:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0 = ((a_n x + a_{n-1})x + \ldots)x + a_0$$

- *Addition/Subtraction* — Adding and subtracting polynomials is even easier than the same operations on long integers, since there is no borrowing or carrying. Simply add or subtract the coefficients of the ith terms for all i from zero to the maximum degree.

Álgebra

Manipulating Polynomial

- *Multiplication* — The product of polynomials $P(x)$ and $Q(x)$ is the sum of the product of every pair of terms, where each term comes from a different polynomial:

$$P(x) \times Q(x) = \sum_{i=0}^{degree(P)} \sum_{j=0}^{degree(Q)} (c_i c_j) x^{i+j}$$

  Such an all-against-all operation is called a *convolution*. Other convolutions in this book include integer multiplication (all digits against all digits) and string matching (all possible positions of the pattern string against all possible text positions). There is an amazing algorithm (the fast Fourier transform, or FFT) which computes convolutions in $O(n \log n)$ time instead of $O(n^2)$, but it is well beyond the scope of this book. Still, it is nice to recognize when you are doing a convolution so you know that such tools exist.

- *Division* — Dividing polynomials is a tricky business, since the polynomials are not closed under division. Note that $1/x$ may or may not be thought of as a polynomial, since it is x^{-1}, but $2x/(x^2+1)$ certainly isn't. It is a *rational function*.

Álgebra

Root Finding

Given a polynomial $P(x)$ and a target number t, the problem of *root finding* is identifying any or all x such that $P(x) = t$.

If $P(x)$ is a first-degree polynomial, the root is simply $x = (t - a_0)/a_1$, where a_i is the coefficient of x_i in $P(x)$. If $P(x)$ is a second-degree polynomial, then the *quadratic equation* applies:

$$x = \frac{-a_1 \pm \sqrt{{a_1}^2 - 4a_2(a_0 - t)}}{2a_2}$$

Álgebra

Root Finding

Given a polynomial $P(x)$ and a target number t, the problem of *root finding* is identifying any or all x such that $P(x) = t$.

If $P(x)$ is a first-degree polynomial, the root is simply $x = (t - a_0)/a_1$, where a_i is the coefficient of x_i in $P(x)$. If $P(x)$ is a second-degree polynomial, then the *quadratic equation* applies:

$$x = \frac{-a_1 \pm \sqrt{{a_1}^2 - 4a_2(a_0 - t)}}{2a_2}$$

Beyond quadratic equations, numerical methods are typically used. Any text on numerical analysis will describe a variety of root-finding algorithms, including Newton's method and Newton-Raphson, as well as many potential traps such as numerical stability. But the basic idea is that of binary search. Suppose a function $f(x)$ is *monotonically increasing* between l and u, meaning that $f(i) \leq f(j)$ for all $l \leq i \leq j \leq u$. Now suppose we want to find the x such that $f(x) = t$. We can compare $f((l + u)/2)$ with t. If $t < f((l + u)/2)$, then the root lies between l and $(l + u)/2$; if not, it lies between $(l+u)/2$ and u. We can keep recurring until the window is narrow enough for our taste.

This method can be used to compute square roots because this is equivalent to solving $x^2 = t$ between 1 and t for all $t \geq 1$. However, a simpler method to find the ith root of t uses exponential functions and logarithms to compute $t^{1/i}$.

Álgebra

Logarithms

$$b^x = y \qquad x = \log_b y.$$

$$\ln x \qquad e = 2.71828\ldots \qquad \exp(\ln x) = x$$

$$\log_a n^b = b \cdot \log_a n$$

$$a^b = \exp(\ln(a^b)) = \exp(b \ln a)$$

$$\log_a b = \frac{\log_c b}{\log_c a}$$

Exemplo

The *Stern-Brocot tree* is a beautiful way for constructing the set of all non-negative fractions $\frac{m}{n}$ where m and n are relatively prime. The idea is to start with two fractions $\left(\frac{0}{1}, \frac{1}{0}\right)$ and then repeat the following operation as many times as desired:

Insert $\frac{m+m'}{n+n'}$ between two adjacent fractions $\frac{m}{n}$ and $\frac{m'}{n'}$.

For example, the first step gives us one new entry between $\frac{0}{1}$ and $\frac{1}{0}$,

$$\frac{0}{1}, \frac{1}{1}, \frac{1}{0}$$

and the next gives two more:

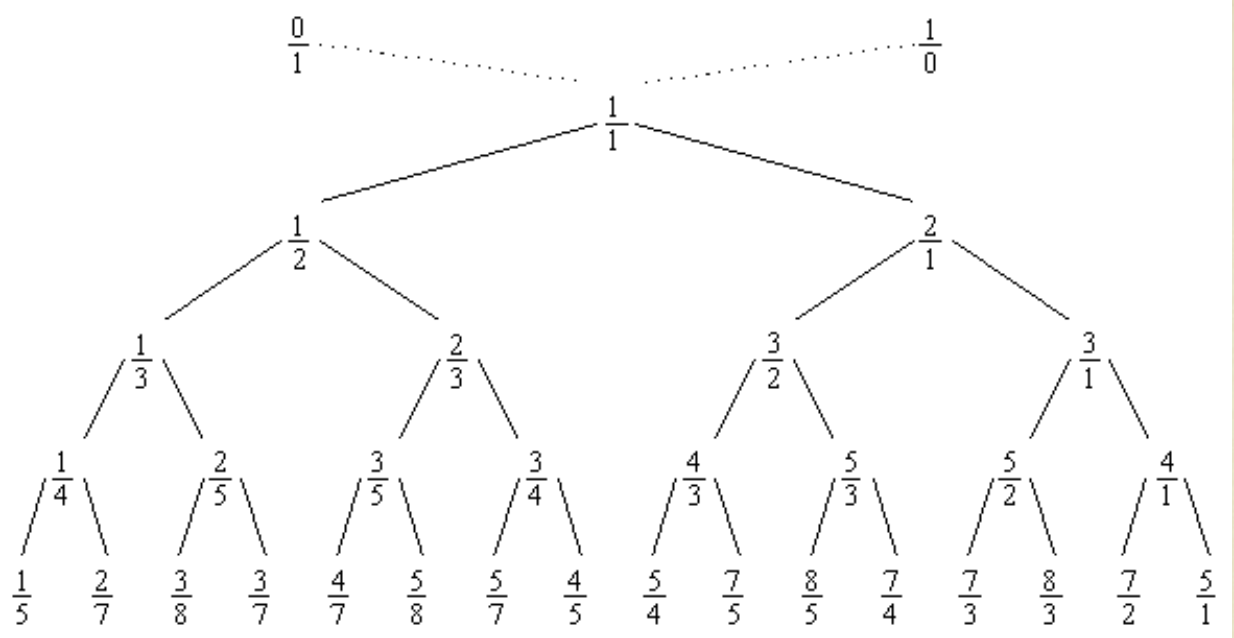
$$\frac{0}{1}, \frac{1}{2}, \frac{1}{1}, \frac{2}{1}, \frac{1}{0}$$

The next gives four more:

$$\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}, \frac{3}{2}, \frac{2}{1}, \frac{3}{1}, \frac{1}{0}$$

The entire array can be regarded as an infinite binary tree structure whose top levels look like this–

Exemplo



This construction preserves order, and thus we cannot possibly get the same fraction in two different places.

We can, in fact, regard the *Stern-Brocot tree* as a *number system* for representing rational numbers, because each positive, reduced fraction occurs exactly once. Let us use the letters "`L`" and "`R`" to stand for going down the left or right branch as we proceed from the root of the tree to a particular fraction; then a string of `L`'s and `R`'s uniquely identifies a place in the tree. For example, `LRRL` means that we go left from $\frac{1}{1}$ down to $\frac{1}{2}$, then right to $\frac{2}{3}$, then right to $\frac{3}{4}$, then left to $\frac{5}{7}$. We can consider LRRL to be a representation of $\frac{5}{7}$. Every positive fraction gets represented in this way as a unique string of `L`'s and `R`'s.

Exemplo

Well, almost every fraction. The fraction $\frac{1}{1}$ corresponds to the empty string. We will denote it by I, since that looks something like 1 and stands for "identity."

In this problem, given a positive rational fraction, represent it in the *Stern-Brocot number system*.

Input

The input file contains multiple test cases. Each test case consists of a line containing two positive integers m and n, where m and n are relatively prime. The input terminates with a test case containing two 1's for m and n, and this case must not be processed.

Output

For each test case in the input file, output a line containing the representation of the given fraction in the *Stern-Brocot number system*.

Sample Input

```
5 7
878 323
1 1
```

Sample Output

```
LRRL
RRLRRLRLLLLRLRRR
```

```cpp
#include <cstdlib>
#include <iostream>
using namespace std;
void calcula(int n, int m){
    int ln=0;
    int lm=1;
    int mn=1;
    int mm=1;
    int rn=1;
    int rm=0;
    double f, mf;
    f=float(n)/float(m);

    while(mn!=n || mm!=m){
        mf=float(mn)/float(mm);
        if(f<mf){//pega esquerda (L)
            rn=mn;
            rm=mm;
            mn+=ln;
            mm+=lm;
            cout << "L";
        }else{//pega direita (R)
            ln=mn;
            lm=mm;
            mn+=rn;
            mm+=rm;
            cout << "R";
        }
    }
    cout << endl;
}

int main(int argc, char *argv[]) {
    int n, m;
    while(cin >> n >> m){
        if(n==1 && m==1) break;
        calcula(n,m);
    }
    return 0;
}
```

Técnicas de Contagem

- *Product Rule* — The *product rule* states that if there are $|A|$ possibilities from set A and $|B|$ possibilities from set B, then there are $|A| \times |B|$ ways to combine one from A and one from B. For example, suppose you own 5 shirts and 4 pants. Then there are $5 \times 4 = 20$ different ways you can get dressed tomorrow.

- *Sum Rule* — The *sum rule* states that if there are $|A|$ possibilities from set A and $|B|$ possibilities from set B, then there are $|A| + |B|$ ways for either A *or* B to occur – assuming the elements of A and B are distinct. For example, given that you own 5 shirts and 4 pants and the laundry ruined one of them, there are 9 possible ruined items.[1]

Técnicas de Contagem

- *Inclusion-Exclusion Formula* — The sum rule is a special case of a more general formula when the two sets can overlap, namely,

$$|A \cup B| = |A| + |B| - |A \cap B|$$

For example, let A represent the set of colors of my shirts and B the colors of my pants. Via inclusion-exclusion, I can calculate the total number of colors given the number of color-matched garments or vice versa. The reason this works is that summing the sets double counts certain possibilities, namely, those occurring in both sets. The inclusion-exclusion formula generalizes to three sets and beyond in a natural way:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Técnicas de Contagem

- *Permutations* — A *permutation* is an arrangement of n items, where every item appears exactly once. There are $n! = \prod_{i=1}^{n} i$ different permutations. The $3! = 6$ permutations of three items are 123, 132, 213, 231, 312, and 321. For $n = 10$, $n! = 3,628,800$, so we start to approach the limits of exhaustive search.
- *Subsets* — A *subset* is a selection of elements from n possible items. There are 2^n distinct subsets of n things. Thus there are $2^3 = 8$ subsets of three items, namely, 1, 2, 3, 12, 13, 23, 123, and the empty set: never forget the empty set. For $n = 20$, $2^n = 1,048,576$, so we start to approach the limits of exhaustive search.
- *Strings* — A *string* is a sequence of items which are drawn *with repetition*. There are m^n distinct sequences of n items drawn from m items. The 27 length-3 strings on 123 are 111, 112, 113, 121, 122, 123, 131, 132, 133, 211, 212, 213, 221, 222, 223, 231, 232, 233, 311, 312, 313, 321, 322, 323, 331, 332, and 333. The number of binary strings of length n is identical to the number of subsets of n items (why?), and the number of possibilities increases even more rapidly with larger m.

Relações de Recorrência

$$a_n = a_{n-1} + 1, a_1 = 1 \longrightarrow a_n = n$$

$$a_n = 2a_{n-1}, a_1 = 2 \longrightarrow a_n = 2^n$$

$$a_n = na_{n-1}, a_1 = 1 \longrightarrow a_n = n!$$

Coeficientes Binomiais

$$\binom{n}{k} = \frac{n \cdot (n-1) \cdots (n-k+1)}{k \cdot (k-1) \cdots 1} = \frac{n!}{k!(n-k)!} \quad \text{if } n \geq k \geq 0$$

- *Committees* — How many ways are there to form a k-member committee from n people? By definition, $\binom{n}{k}$ is the answer.
- *Paths Across a Grid* — How many ways are there to travel from the upper-left corner of an $n \times m$ grid to the lower-right corner by walking only down and to the right? Every path must consist of $n + m$ steps, n downward and m to the right. Every path with a different set of downward moves is distinct, so there are $\binom{n+m}{n}$ such sets/paths.
- *Coefficients of* $(a+b)^n$ — Observe that

$$(a+b)^3 = 1a^3 + 3a^2b + 3ab^2 + 1b^3$$

What is the coefficient of the term $a^k b^{n-k}$? Clearly $\binom{n}{k}$, because it counts the number of ways we can choose the k a-terms out of n possibilities.

Triângulo de Pascal

```
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5  10  10   5   1
```

$(n+1)$st row of the table gives the values $\binom{n}{i}$ for $0 \leq i \leq n$.

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Exemplo

How many different ordered triples (a,b,c) of non-negative integers are there such a+b+c=50 ?

Exemplo

How many different ordered triples (a,b,c) of non-negative integers are there such a+b+c=50 ?

a+b=n n+1 solutions

(0,n)(1,n-1),(2,n-2),...,(n,0)

Exemplo

How many different ordered triples (a,b,c) of non-negative integers are there such a+b+c=50 ?

a+b=n n+1 solutions

(0,n)(1,n-1),(2,n-2),...,(n,0)

c=0, a+b=50
c=1, a+b=49
c=2, a+b=48 1+2+3+...+51 = (51*52)/2 = C(52,2)
...
c=37, a+b=33

Exemplo

How many different ordered triples (a,b,c) of non-negative integers are there such a+b+c=11 ?

C(13,2)

How many different ordered triples (a,b,c) of non-negative integers are there such a+b+c=50 ?

C(52,2)

Exemplo

How many different ways we can place b indistinguishable into u distinguishable urns ?

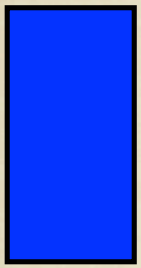
$$C(b+u-1,b) = C(b+u-1, u-1)$$

$$B=50\ U=3$$

$$C(52, 2)$$

Outras Seqüências

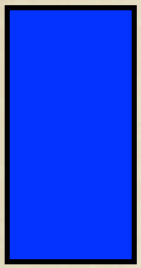
Define a domino to be a 1x2 rectangle. In how many ways can an nx2 rectanle be tiled by dominos ?



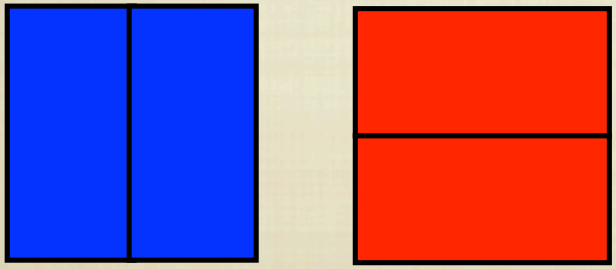
$T_1 = 1$

Outras Seqüências

Define a domino to be a 1x2 rectangle. In how many ways can an nx2 rectanle be tiled by dominos ?



$T_1 = 1$

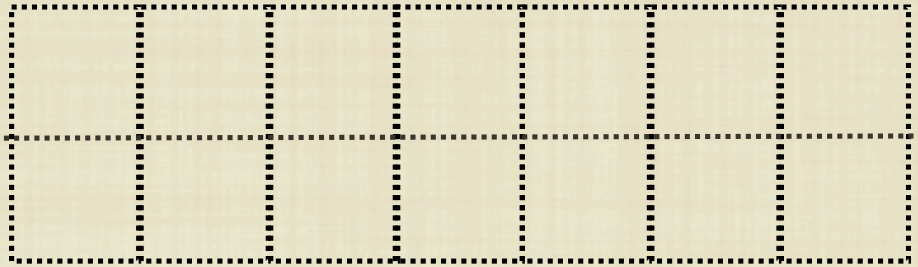


$T_2 = 2$

Outras Seqüências

Define a domino to be a 1x2 rectangle. In how many ways can an nx2 rectanle be tiled by dominos ?

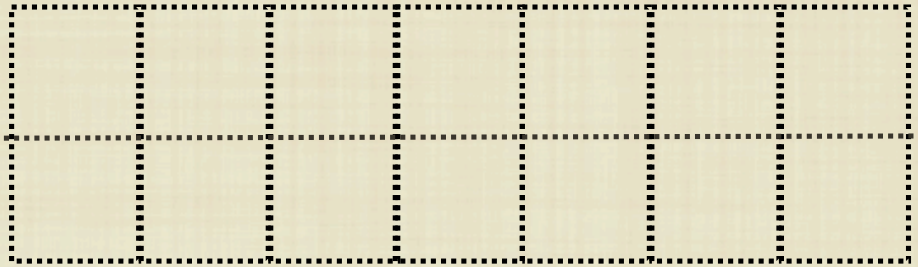
$T_7 = ?$

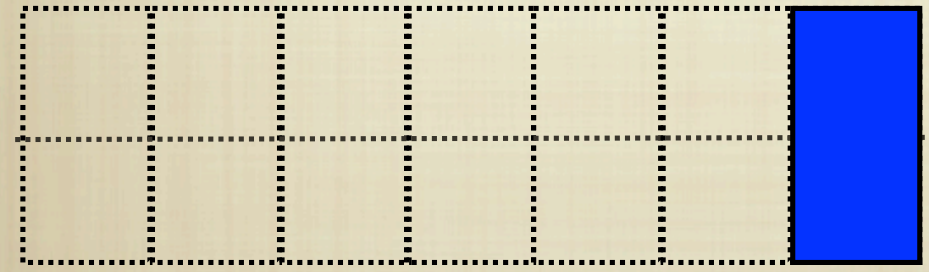


Outras Seqüências

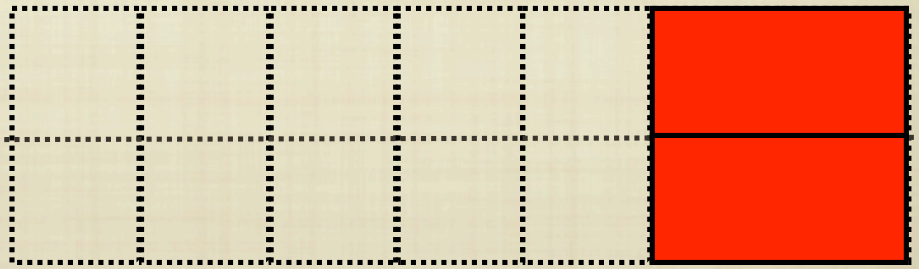
Define a domino to be a 1x2 rectangle. In how many ways can an nx2 rectanle be tiled by dominos ?

$T_7 = ?$





6



5

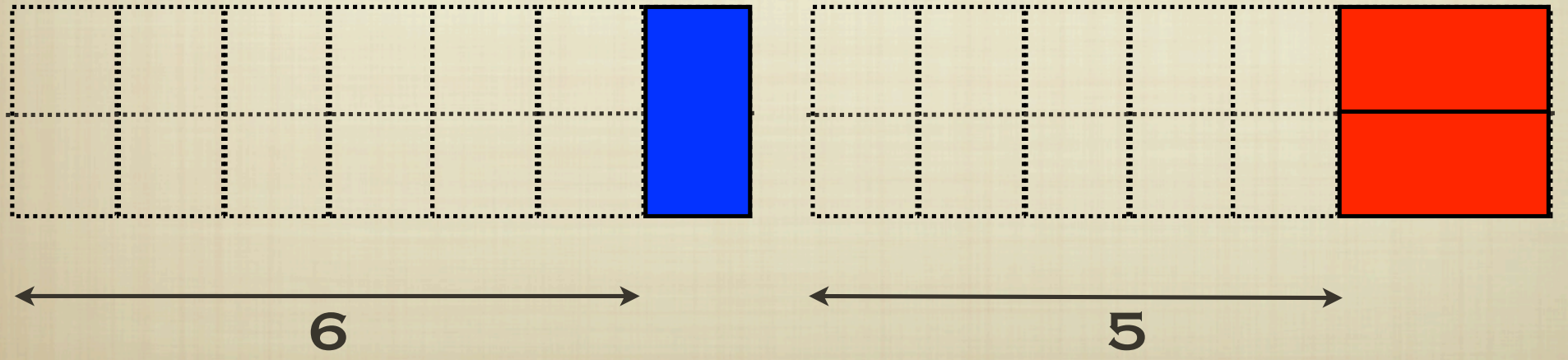
Outras Seqüências

Define a domino to be a 1x2 rectangle. In how many ways can an nx2 rectanle be tiled by dominos ?

$T_7 = T_6 + T_5$

$T_{N+1} = T_N + T_{N-1}$

$T_1 = 1, T_2 = 2$



Outras Seqüências

3 numbers to make a proportion

$$\frac{a}{b} = \frac{b}{c}$$

2 numbers to make a proportion

$$\frac{a}{b} = \frac{b}{a+b}$$

$$x = \frac{1 \pm \sqrt{5}}{2}$$

$$\left(\frac{b}{a}\right)^2 = \left(\frac{b}{a}\right) + 1$$

$$GoldenRatio\ \varphi = \frac{1 + \sqrt{5}}{2}$$

$$x^2 - x - 1 = 0$$

Outras Seqüências

$$
\begin{aligned}
x &= \lim_{n\to\infty} \frac{F(n+1)}{F(n)} \\
&= \lim_{n\to\infty} \frac{F(n)+F(n-1)}{F(n)} \\
&= \lim_{n\to\infty} \left(\frac{F(n)}{F(n)} + \frac{F(n-1)}{F(n)} \right) \\
&= 1 + \lim_{n\to\infty} \frac{F(n-1)}{F(n)} \\
&= 1 + \frac{1}{\lim_{n\to\infty} \frac{F(n)}{F(n-1)}} \\
&= 1 + \frac{1}{x}
\end{aligned}
$$

$$\frac{x}{1} = \frac{1}{x-1}$$

Johannes Kepler

Outras Seqüências

- *Fibonacci numbers* — Defined by the recurrence $F_n = F_{n-1} + F_{n-2}$ and the initial values $F_0 = 0$ and $F_1 = 1$, they emerge repeatedly because this is perhaps the simplest interesting recurrence relation. The first several values are 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... The Fibonacci numbers lend themselves to an amazing variety of mathematical identities, and are just fun to play with. They have the following hard-to-guess but simple-to-derive closed form:

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

  This closed form has certain important implications. Since $(1-\sqrt{5})/2$ is between 0 and 1, raising it to any power leaves a number in this range. Thus the first term, ϕ^n where $\phi = (1+\sqrt{5})/2$ is the driving quantity, and can be used to estimate F_n to within plus or minus 1.

Outras Seqüências

- How many ways are there to build a balanced formula from n sets of left and right parenthesis ?
  - for n = 3 ?

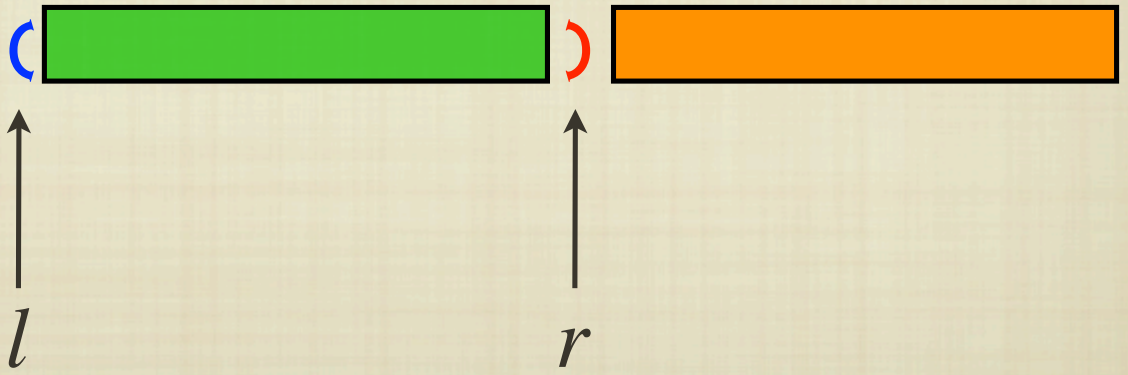
Outras Seqüências

- How many ways are there to build a balanced formula from n sets of left and right parenthesis ?
  - for n = 3 ? there are 5 different ways:
    - ((()))
    - ()(())
    - (())()
    - (()())
    - ()()()

Outras Seqüências

- How many ways are there to build a balanced formula from n sets of left and right parenthesis ?

  - for n = 3 ? there are 5 different ways:
    - ((()))
    - ()(())
    - (())()
    - (()())
    - ()()()

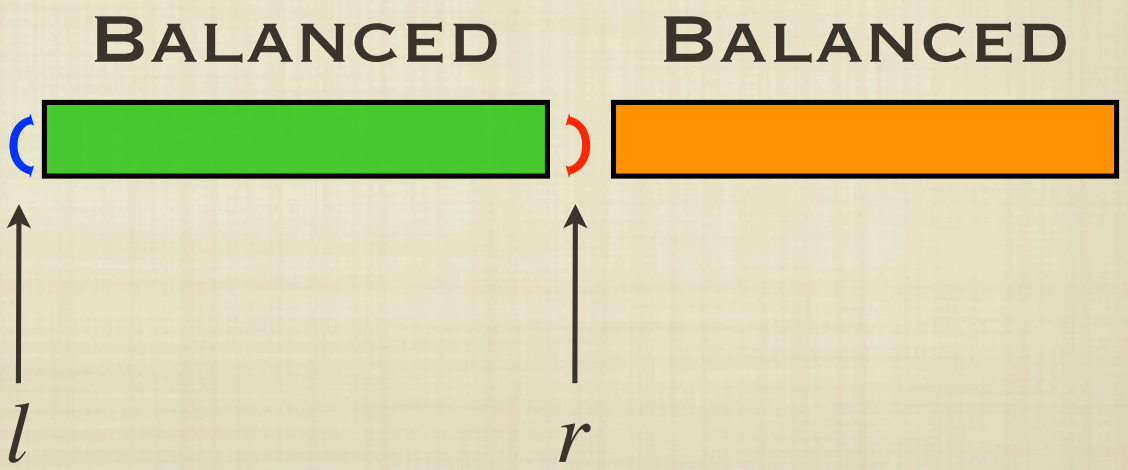


lefmost

Outras Seqüências

- How many ways are there to build a balanced formula from n sets of left and right parenthesis ?

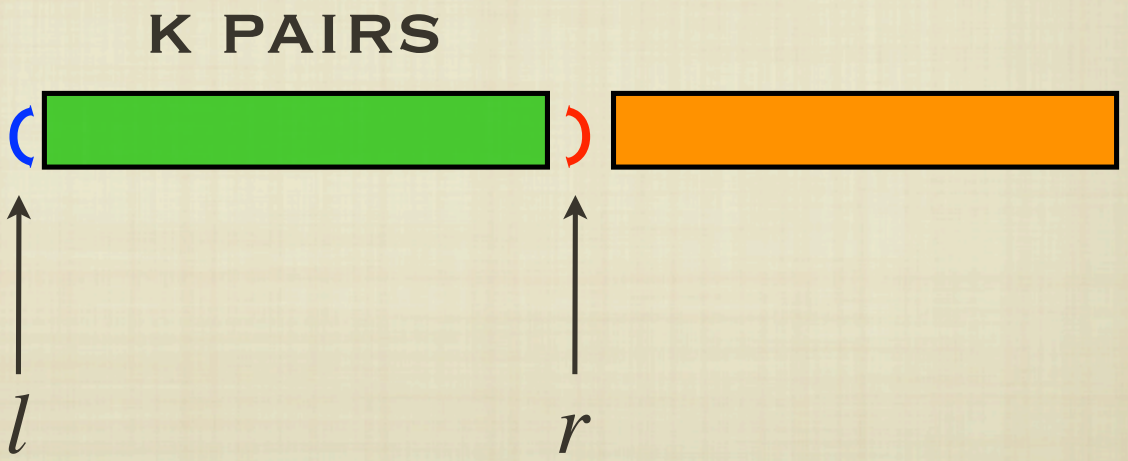
  - for n = 3 ? there are 5 different ways:
    - ((()))
    - ()(())
    - (()())
    - (())()
    - ()()()



Outras Seqüências

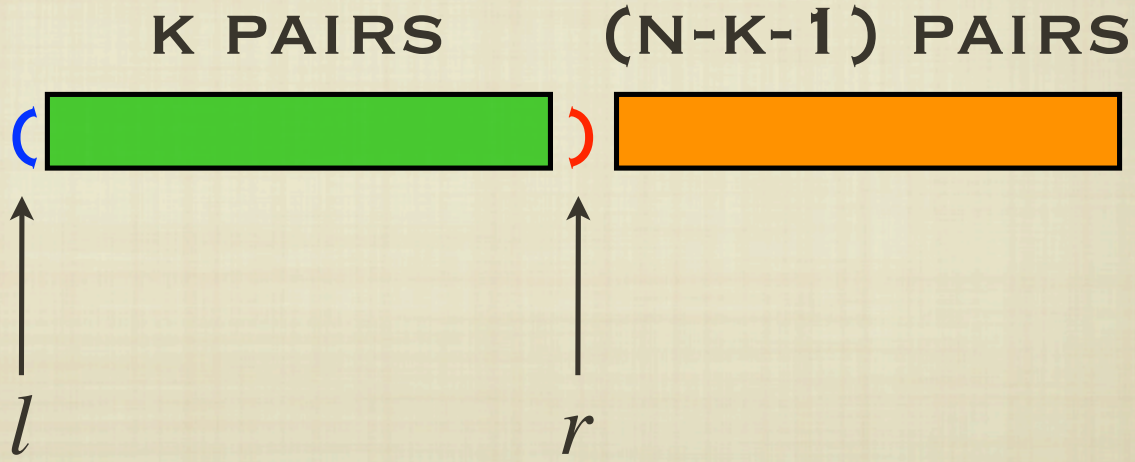
- How many ways are there to build a balanced formula from n sets of left and right parenthesis ?

  - for n = 3 ? there are 5 different ways:
    - ((()))
    - ()(())
    - (())()
    - (()())
    - ()()()



Outras Seqüências

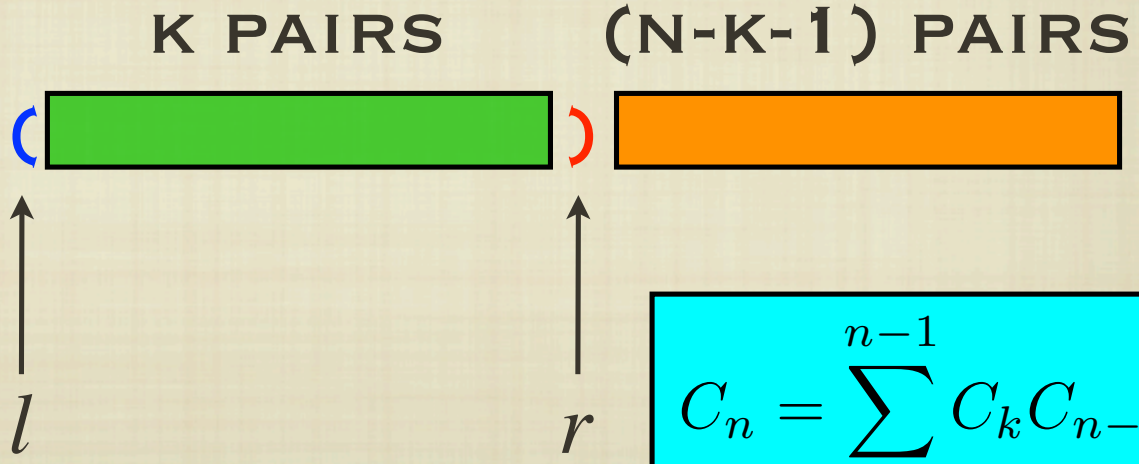
- How many ways are there to build a balanced formula from n sets of left and right parenthesis ?
  - for n = 3 ? there are 5 different ways:
    - ((()))
    - ()(())
    - (())()
    - (()())
    - ()()()



Outras Seqüências

- How many ways are there to build a balanced formula from n sets of left and right parenthesis ?

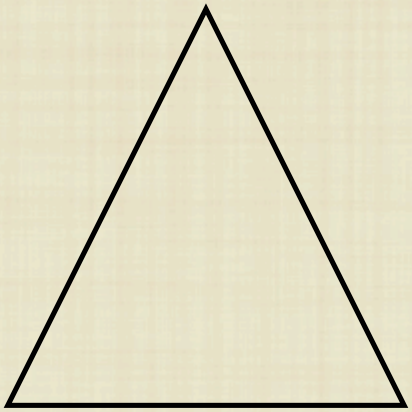
- for n = 3 ? there are 5 different ways:
  - ((()))
  - ()(())
  - (())()
  - (()())



$$C_n = \sum_{k=0}^{n-1} C_k C_{n-1-k}$$

Outras Seqüências

- How many ways to triangulate a convex polygon ?



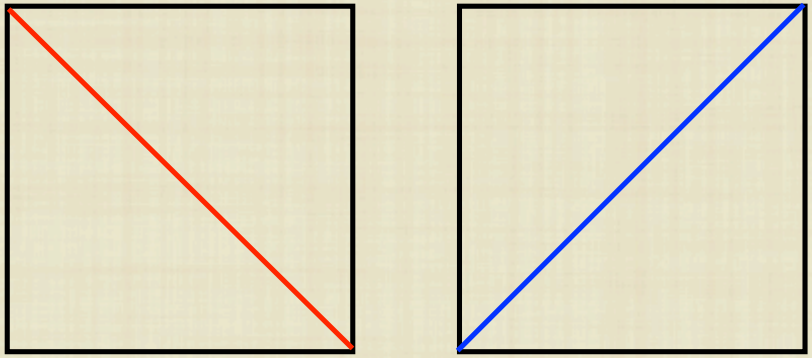
$T_3 = 1$

Outras Seqüências

- How many ways to triangulate a convex polygon ?

Outras Seqüências

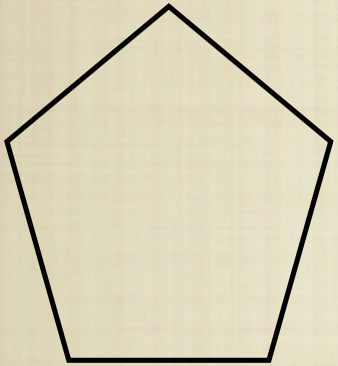
- How many ways to triangulate a convex polygon ?



$T_4 = 2$

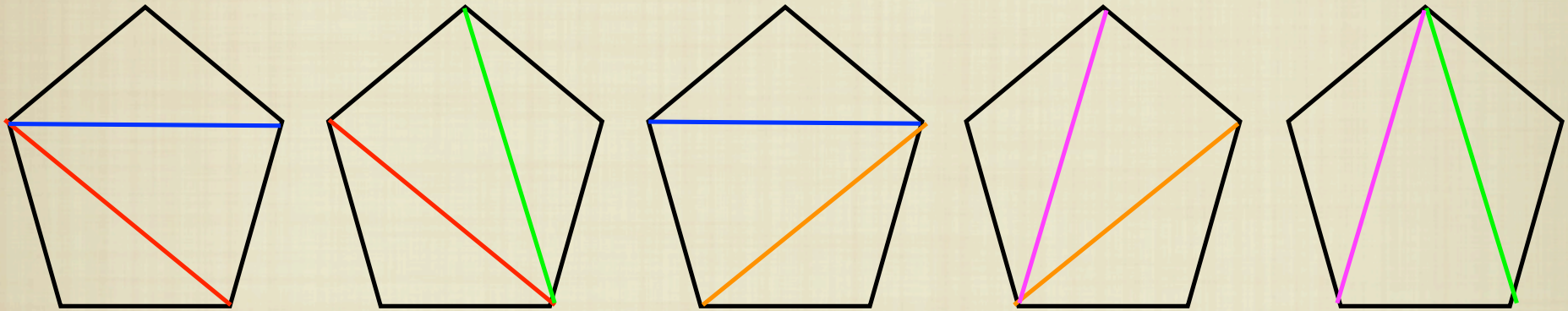
Outras Seqüências

- How many ways to triangulate a convex polygon ?



Outras Seqüências

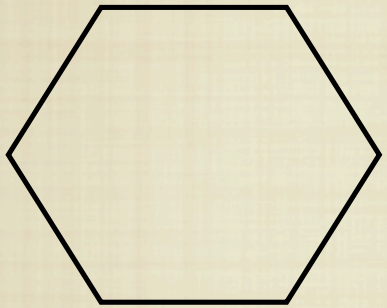
- How many ways to triangulate a convex polygon ?



$T_4 = 5$

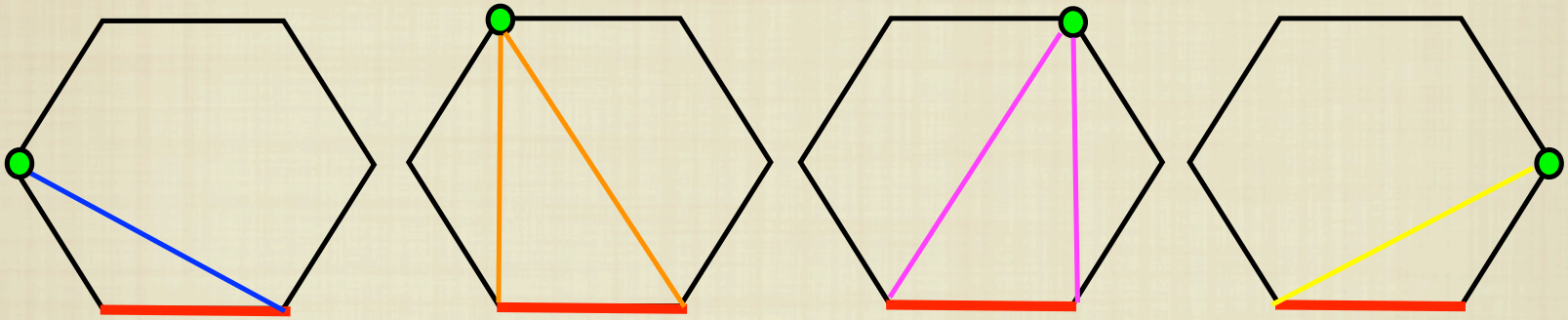
Outras Seqüências

- How many ways to triangulate a convex polygon ?



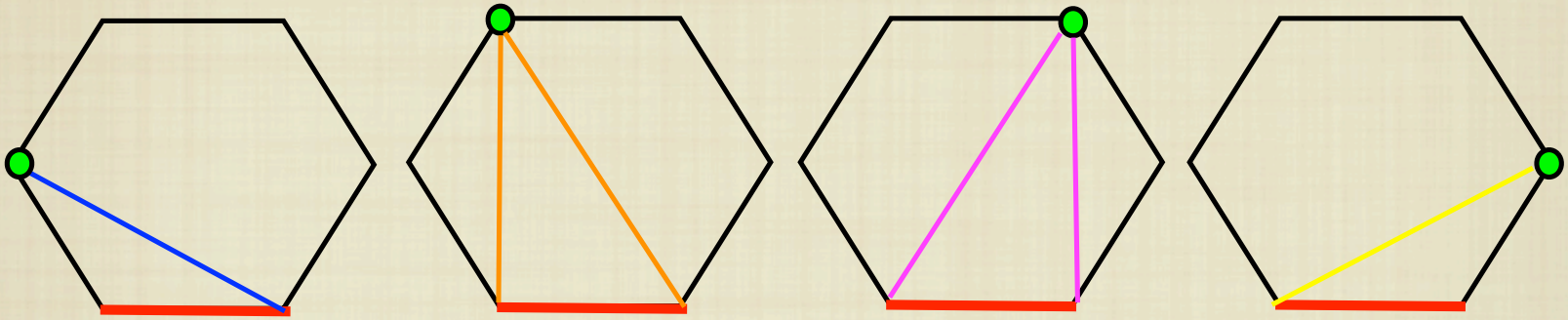
Outras Seqüências

- How many ways to triangulate a convex polygon ?



Outras Seqüências

- How many ways to triangulate a convex polygon ?



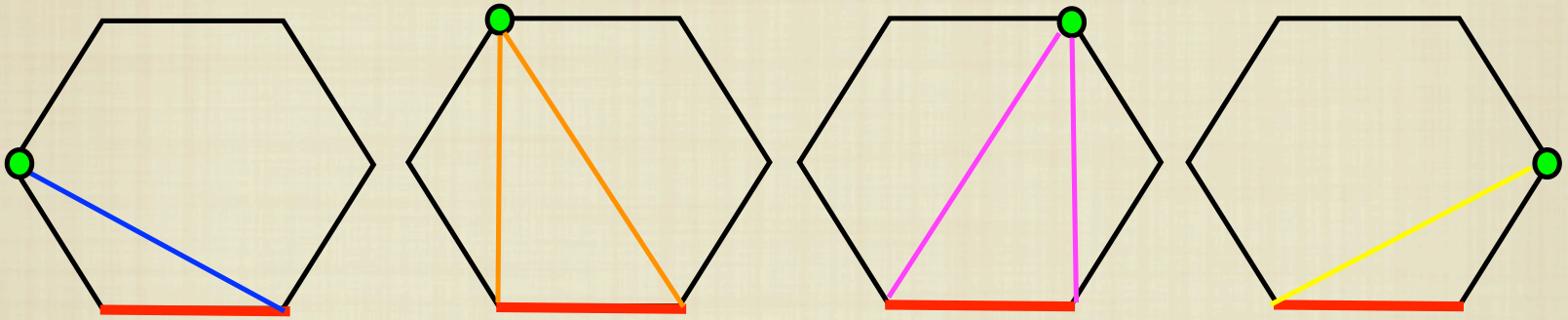
$$T_6 = T_3 * T_5 + T_3 * T_4 + T_4 * T_3 + T_5 * T_3$$

Outras Seqüências

- How many ways to triangulate a convex polygon ?

$$C_n = \sum_{k=0}^{n-1} C_k C_{n-1-k}$$

$T_4 = C_2$
$T_5 = C_3$
$T_6 = C_4$
$C_0 = 1$



$$T_6 = T_3 * T_5 + T_3 * T_4 + T_4 * T_3 + T_5 * T_3$$

$$T_6 = 1*5 + 1*2 + 2*1 + 5*1 = 14$$

Outras Seqüências

- *Catalan Numbers* — The recurrence and associated closed form

$$C_n = \sum_{k=0}^{n-1} C_k C_{n-1-k} = \frac{1}{n+1}\binom{2n}{n}$$

defines the *Catalan numbers*, which occur in a surprising number of problems in combinatorics. The first several terms are $2, 5, 14, 42, 132, 429, 1430, \ldots$ when $C_0 = 1$.

Outras Seqüências

- *Eulerian Numbers* — The *Eulerian* numbers $\left\langle {n \atop k} \right\rangle$ count the number of permutations of length n with exactly k ascending sequences or *runs*. A recurrence can be formulated by considering each permutation p of $1, ..., n-1$. There are n places to insert element n, and each either splits an existing run of p or occurs immediately after the last element of an existing run, thus preserving the run count. Thus $\left\langle {n \atop k} \right\rangle = k\left\langle {n-1 \atop k} \right\rangle + (n-k+1)\left\langle {n-1 \atop k-1} \right\rangle$. Can you construct the eleven permutations of length four with exactly two runs?

Outras Seqüências

- *Eulerian Numbers* — The *Eulerian* numbers $\left\langle {n \atop k} \right\rangle$ count the number of permutations of length n with exactly k ascending sequences or *runs*. A recurrence can be formulated by considering each permutation p of $1, ..., n-1$. There are n places to insert element n, and each either splits an existing run of p or occurs immediately after the last element of an existing run, thus preserving the run count. Thus $\left\langle {n \atop k} \right\rangle = k\left\langle {n-1 \atop k} \right\rangle + (n-k+1)\left\langle {n-1 \atop k-1} \right\rangle$. Can you construct the eleven permutations of length four with exactly two runs?

1324-1423-2314-2413-3412

1243-1342-2341-2134-3124-4123

Outras Seqüências

- *Stirling Numbers* — There are two different types of Stirling numbers. The first, $\begin{bmatrix} n \\ k \end{bmatrix}$, counts the number of permutations on n elements with exactly k cycles. To formulate the recurrence, observe the nth element either forms a singleton cycle or it doesn't. If it does, there are $\begin{bmatrix} n-1 \\ k-1 \end{bmatrix}$ ways to arrange the rest of the elements to form $k-1$ cycles. If not, the nth element can be inserted in every possible position of every cycle of the $\begin{bmatrix} n-1 \\ k \end{bmatrix}$ ways to make k cycles out of $n-1$ elements. Thus

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n-1 \\ k-1 \end{bmatrix} + (n-1)\begin{bmatrix} n-1 \\ k \end{bmatrix}$$

  There are 11 permutations of four elements with exactly two cycles.

Outras Seqüências

- *Stirling Numbers* — There are two different types of Stirling numbers. The first, $\begin{bmatrix} n \\ k \end{bmatrix}$, counts the number of permutations on n elements with exactly k cycles. To formulate the recurrence, observe the nth element either forms a singleton cycle or it doesn't. If it does, there are $\begin{bmatrix} n-1 \\ k-1 \end{bmatrix}$ ways to arrange the rest of the elements to form $k-1$ cycles. If not, the nth element can be inserted in every possible position of every cycle of the $\begin{bmatrix} n-1 \\ k \end{bmatrix}$ ways to make k cycles out of $n-1$ elements. Thus

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n-1 \\ k-1 \end{bmatrix} + (n-1)\begin{bmatrix} n-1 \\ k \end{bmatrix}$$

  There are 11 permutations of four elements with exactly two cycles.

[1,2,3][4] [1,2,4][3] [1,3,4][2] [2,3,4][1]
[1,3,2][4] [1,4,2][3] [1,4,3][2] [2,4,3][1]
[1,2][3,4] [1,3][2,4] [1,4][2,3]

Outras Seqüências

- *Set Partitions* — The second kind of Stirling number $\left\{ {n \atop k} \right\}$ counts the number of ways to partition n items into k sets. For example, there are seven ways to partition four items into exactly two subsets: (1)(234), (12)(34), (13)(24), (14)(23), (123)(4), (124)(3) and (134)(2). The nth item can be inserted into any of the k subsets of an $n-1$-part partition or it forms a singleton set. Thus by a similar argument to that of the other Stirling numbers they are defined by the recurrence $\left\{ {n \atop k} \right\} = k\left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k} \right\}$. The special case of $\left\{ {n \atop 2} \right\} = 2^{n-1} - 1$, since any proper subset of the elements 2 to n can be unioned with (1) to define the set partition. The second part of the partition consists of exactly the elements not in this first part.

n \ k	0	1	2	3	4	5	6	7	8	9
0	1									
1	0	1								
2	0	1	1							
3	0	1	3	1						
4	0	1	7	6	1					
5	0	1	15	25	10	1				
6	0	1	31	90	65	15	1			
7	0	1	63	301	350	140	21	1		
8	0	1	127	966	1701	1050	266	28	1	
9	0	1	255	3025	7770	6951	2646	462	36	1

Outras Seqüências

- *Integer Partitions* — An integer partition of n is an unordered set of positive integers which add up to n. For example, there are seven partitions of 5, namely, (5), $(4,1)$, $(3,2)$, $(3,1,1)$, $(2,2,1)$, $(2,1,1,1)$, and $(1,1,1,1,1)$. The easiest way to count them is to define a function $f(n,k)$ giving the number of integer partitions of n with largest part at most k. In any acceptable partition the largest part either does or does not reach with limit, so $f(n,k) = f(n-k,k) + f(n,k-1)$. The basis cases are $f(1,1) = 1$ and $f(n,k) = 0$ whenever $k > n$.

Recursão e Indução

For example, consider the following recurrence relation:

$$T_n = 2T_{n-1} + 1, T_0 = 0$$

Building a table of values yields the following:

n	0	1	2	3	4	5	6	7
T_n	0	1	3	7	15	31	63	127

Can you guess what the solution is? You should notice that things look like they are doubling each time, no surprise considering the formula. But it is not quite 2^n. By playing around with variations of this function, you should be able to stumble on the conjecture that $T_n = 2^n - 1$. To finish the job, we must prove this conjecture, using the three steps of induction:

1. Show that the basis is true: $T_0 = 2^0 - 1 = 0$.
2. Now assume it is true for T_{n-1}.
3. Use this assumption to complete the argument:

$$T_n = 2T_{n-1} + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 1$$

Exponentiation by squaring

$$x^n = \begin{cases} 1, & \text{if } n = 0 \\ \frac{1}{x}^{-n}, & \text{if } n < 0 \\ x \cdot \left(x^{\frac{n-1}{2}}\right)^2, & \text{if } n \text{ is odd} \\ \left(x^{\frac{n}{2}}\right)^2, & \text{if } n \text{ is even} \end{cases}$$