Teaoria Combinatoria

Permutaciones

**class** Permutation {

**private** Integer[] a;

**private** BigInteger numLeft;

**private** BigInteger total;

**public** Permutation (**int** n) {

a = **new** Integer[n];

total = getFactorial (n);

reset ();

}

**public** **void** reset () {

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

a[i] = i+1;

}

numLeft = **new** BigInteger (total.toString ());

}

**public** BigInteger getNumLeft () {// Return number of permutations not yet generated

**return** numLeft;

}

**public** BigInteger getTotal () {// Return total number of permutations

**return** total;

}

**public** **boolean** hasMore () { // Are there more permutations?

**return** numLeft.compareTo (BigInteger.ZERO) == 1;

}

**private** **static** BigInteger getFactorial (**int** n) { // Compute factorial

BigInteger fact = BigInteger.ONE;

**for** (**int** i = n; i > 1; i--) {

fact = fact.multiply (**new** BigInteger (Integer.toString (i)));

}

**return** fact;

}

**public** Integer[] getNext () { // Generate next permutation (algorithm from Rosen p. 284)

**if** (numLeft.equals (total)) {

numLeft = numLeft.subtract (BigInteger.ONE);

**return** a;

}

**int** temp;

// Find largest index j with a[j] < a[j+1]

**int** j = a.length - 2;

**while** (a[j] > a[j+1]) {

j--;

}

// Find index k such that a[k] is smallest integer

// greater than a[j] to the right of a[j]

**int** k = a.length - 1;

**while** (a[j] > a[k]) {

k--;

}

// Interchange a[j] and a[k]

temp = a[k];

a[k] = a[j];

a[j] = temp;

// Put tail end of permutation after jth position in increasing order

**int** r = a.length - 1;

**int** s = j + 1;

**while** (r > s) {

temp = a[s];

a[s] = a[r];

a[r] = temp;

r--;

s++;

}

numLeft = numLeft.subtract (BigInteger.ONE);

**return** a;

}

}

**Olivia**

Algunas sucesiones:

1+2+3+,,,+n = n(n+1)/2

2+4+6+,,,+2n = n(n+1)

1+3+5+,,,+2n-1 = n^2

1^2+2^2+3^2+,,,+n^2+ = n(n+1)(2n+1)/6

1^3+2^3+3^3+,,,+n^3+ = (n(n+1)/2)^2

1^2+3^2+5^2+,,,+2n-1^2 = n(2n-1)(2n+1)/3

1/1\*2+1/2\*3+1/3\*4+,,,+1/n(n+1) = n/(n+1)

1-(1/2^2)+1-(1/3^2)+,,,,+1-(1/n^2) = (n+1)/2n