Permutations

- Permutations on 3 elements:

  <u>a</u>bc

  <u>a</u>cb

  <u>b</u>ac

  <u>b</u>ca

  <u>c</u>ab

  <u>c</u>ba

- N! different permutations on N elements

Permutations

```
Public static void perm(Object [] a, int k, int m)
{ // generate all permutations of list[k:m]
    int j;
    if (k == m) printPerm(list, m);
    else
      for (j=k; j <= m; j++)
      {
          swap(list, k, j);
          perm(list, k+1, m);
          swap(list, k, j);
      }
}
```

Generating Random Permutations

```
public static final void randomPerm(Object [] a)   //p333
{
    Random r = new Random();
    for (int j = 1; j < a.length; j++)
        swap (a, j, r.nextInt(0,j));
}
```

Random Number Generator

- $X_{i+1} = A \, X_i \pmod{M}$
- If M=11, A=7, then the sequence is:
  7, 5, 2, 3, 10, 4, 6, 9, 8, 1, 7, 5, 2, …
- Another <u>good</u> choice is: M = 2,147,483,647, and A = 48,271
- Major problem: <u>OVERFLOW</u> (on a 32-bit machine)
- Improvement:
  Q = M / A;
  R = M % A;
  $X_{i+1} = A \, (X_i \% Q) - R(X_i/Q) + M \, \delta(X_i);$
  // $\delta(X_i)$ is 0 if sum of first 2 terms is +ve, else it is 1