

Random Number Algorithms

shuffle() Algorithm

- `shuffle()` rearranges the elements in a random order. It takes a random number engine as its third argument

```
vector<int> v {3, 1, 4, 1, 5, 9};  
static mt19937 mt;  
shuffle(v.begin(), v.end(), mt);
```

- `shuffle()` calls `uniform_int_distribution` internally
- It gives perfect shuffling - all permutations are equally likely
- `random_shuffle()` uses `rand()` and is deprecated

- `bernoulli_distribution` produces boolean values
- This is useful for making one-off decisions with random outcomes

```
static mt19937 mt;  
static bernoulli_distribution b;  
  
if (b(mt))  
    cout << "Your subjects are grateful for your wise and benevolent rule\n";  
else  
    cout << "The peasants are revolting!\n";
```