2019/6/24 xor shift.hpp

```
#include "../bits/stdc++.h"
2 // XorShift 乱数生成器
3 // verified: https://atcoder.jp/contests/rco-contest-2019-qual/submissions/4237801
 4 struct XorShift
 6
       using result_type = uint32_t;
       result_type w = 123456789, x = 362436069, y = 521288629, z = 88675123;
       XorShift(result_type seed = time(nullptr))
8
9
10
           w = seed:
11
           x = w << 13;
           y = (w >> 9) ^ (x << 6);
12
           z = y \gg 7;
13
14
       static const result_type min() { return 0; }
static const result_type max() { return 0x7FFFFFFF; }
15
16
17
       result_type operator()()
18
           result_type t = x ^ (x << 11);
19
20
           x = y;
           y = z;
z = w;
21
22
23
           return w = (w ^ (w >> 19) ^ (t ^ (t >> 8)));
24
       }
25
       result_type rand()
26
       {
27
           result_type t = x ^ (x << 11);
28
           x = y;
29
           y = z;
30
           z = w;
           return w = (w ^ (w >> 19) ^ (t ^ (t >> 8)));
31
       }
// [min,max] の整数値乱数
32
33
34
       result_type randInt(result_type min = 0, result_type max = 0x7FFFFFFFF)
35
       {
36
           return rand() % (max - min + 1) + min;
37
38
       ·// [min,max] の浮動小数点乱数
39
       double randDouble(double min = 0, double max = 1)
40
           return (double)(rand() % 0xFFFF) / 0xFFFF * (max - min) + min;
41
42
       ,
// 変数をデフォルト値に設定する
43
       void SetDefault()
44
45
       {
           w = 123456789;
46
47
           x = 362436069;
           y = 521288629;
49
           z = 88675123;
50
51 };
```

localhost:4649/?mode=clike 1/1