

Collecting complete sets

Finding the waiting time in order to collect a "complete set" by simulation.

Reference: Ziegenbalg J.: Algorithmen von Hammurapi bis Gödel,
4. Auflage, Springer-Spektrum, Wiesbaden 2016, Section 4.5.2

1 *Some tests with list operations*

```
(%i1) L : makelist(j*j, j, 0, 6);
```

```
(L) [0,1,4,9,16,25,36]
```

```
(%i2) L;
```

```
(%o2) [0,1,4,9,16,25,36]
```

```
(%i3) L[4];
```

```
(%o3) 9
```

```
(%i4) apply("+", L);
```

```
(%o4) 91
```

```
(%i5) rest(L);
```

```
(%o5) [1,4,9,16,25,36]
```

2 *Some tests with the random generator*

The commands `make_random_state` and `set_random_state` serve the purpose of initializing the random number generator; cf. Maxima's help system.

```
(%i7) s1: make_random_state (654321) $
```

```
set_random_state (s1);
```

```
(%o7) done
```

```
(%i12) make_random_state(true) $
```

```
random(1000);
```

```
L2 : []; for i:1 thru 12 do L2 : append(L2, [random(1000)]); L2;
```

```
(%o9) 768
```

```
(L2) []
```

```
(%o11) done
```

```
(%o12) [800,830,591,447,667,498,958,322,597,110,344,9]
```

3 *The collectors problem*

3.1 Implementation with array

```
(%i22) collector() :=
  block([collector_array, r, i : 0 ],
    make_random_state(true),
    collector_array : make_array(fixnum, 7),
    fillarray(collector_array, makelist(0, j, 0, 6)),
    /* fill collector_array with zeroes */
    if verbose then print(i, rest(listarray(collector_array))),
    while is(apply("*", rest(listarray(collector_array))) = 0)
    do (i : i+1 ,
      r : random(6)+1 ,
      collector_array[r] : collector_array[r]+1 ,
      if verbose then print(i, rest(listarray(collector_array))) ) ,
    [i, rest(listarray(collector_array))] ) $ ;
```

```
(%i23) (verbose : true, collector() );
```

```
0 [0,0,0,0,0,0]
1 [0,0,0,1,0,0]
2 [0,0,0,1,1,0]
3 [0,0,1,1,1,0]
4 [0,1,1,1,1,0]
5 [0,1,2,1,1,0]
6 [0,1,3,1,1,0]
7 [0,1,4,1,1,0]
8 [0,1,4,2,1,0]
9 [0,1,4,2,2,0]
10 [0,2,4,2,2,0]
11 [0,2,5,2,2,0]
12 [0,2,5,2,2,1]
13 [0,2,5,2,3,1]
14 [0,3,5,2,3,1]
15 [0,3,5,3,3,1]
16 [0,3,6,3,3,1]
17 [0,3,6,3,3,2]
18 [0,3,7,3,3,2]
19 [1,3,7,3,3,2]
```

```
(%o23) [19,[1,3,7,3,3,2]]
```

```
(%i24) collector_array_series(n) :=
```

```
  block(L : [],
    for i:1 thru n do L : append(L, [collector()[1]]),
    [L, float(apply("+",L)/length(L))] ) $ ;
```

```
(%i44) (verbose : false, collector_array_series(100) );
```

```
(%o44) [[20,8,7,27,14,12,11,19,35,22,18,21,16,11,20,30,10,16,12,16,16,
14,11,14,15,8,14,19,15,9,11,16,8,28,18,19,9,14,17,15,14,13,21,17,10,14,
,10,10,15,23,9,46,14,13,24,22,15,11,21,14,23,26,16,12,13,14,34,9,9,14,
23,10,16,8,20,20,17,22,11,10,22,13,12,14,17,13,11,9,12,9,11,22,9,10,20
,9,15,10,15,11],15.62]
```

3.2 Implementation purely with list

```
(%i29) collector_list() :=
  block([SL : [0,0,0,0,0,0], r, i : 0 ],
    make_random_state(true),
    if verbose then print(SL),
    while is(apply("*", SL) = 0)
    do (i : i+1,
      r : random(6)+1 ,
      SL[r] : SL[r]+1,
      if verbose then print(SL) ),
    [i, SL] ) $ ;
```

```
(%i30) verbose : true;
```

```
(verbose) true
```

```
(%i31) collector_list();
  [0,0,0,0,0,0]
  [1,0,0,0,0,0]
  [1,0,0,1,0,0]
  [1,0,1,1,0,0]
  [1,1,1,1,0,0]
  [1,1,1,2,0,0]
  [2,1,1,2,0,0]
  [2,1,1,3,0,0]
  [2,1,1,4,0,0]
  [3,1,1,4,0,0]
  [3,1,1,5,0,0]
  [3,1,2,5,0,0]
  [3,1,3,5,0,0]
  [3,1,3,6,0,0]
  [3,1,3,6,0,1]
  [3,1,3,6,1,1]
(%o31) [15,[3,1,3,6,1,1]]
```

4 Some tests with the list operation "apply"

```
→ apply("*", [1,2,3,4]);
```

```
(%o21) 24
```

```
→ apply("*", []);
```

```
(%o22) 1
```

```
(%i32) collector_list_series(n) :=
  block(L : [],
    for i:1 thru n do L : append(L, [collector_list()[1]]),
    [L, float(apply("+",L)/length(L))] ) $ ;
```

```
(%i45) (verbose : false, collector_list_series(100) );
```

```
(%o45) [[15,18,16,34,9,22,9,10,16,44,19,9,10,10,17,9,31,22,35,13,11,8,9,
17,9,9,14,13,31,13,19,19,14,19,11,8,20,25,19,8,20,20,12,10,15,17,14,30
,10,15,10,23,9,22,21,24,10,9,11,21,33,20,6,13,17,11,18,18,24,10,10,19,
14,16,11,8,17,15,17,26,12,24,11,26,11,11,11,15,14,12,13,35,28,8,19,9,8
,9,9,12],16.07]
```

```
(%i41) timer(all);
```

```
(%o41) []
```

```
(%i42) timer;
```

```
(%o42) [collector_list_series,collector_list,collector_array_series,collector]
```

```
(%i46) timer_info();
```

```
(%o46)


| function               | time/call                             | calls | runtime   | gctime |
|------------------------|---------------------------------------|-------|-----------|--------|
| collector_list_series  | 0.016 sec                             | 1     | 0.016 sec | 0      |
| collector_list         | $1.6 \cdot 10^{-4}$ sec               | 100   | 0.016 sec | 0      |
| collector_array_series | 0.015 sec                             | 1     | 0.015 sec | 0      |
| collector              | $1.5 \cdot 10^{-4}$ sec               | 100   | 0.015 sec | 0      |
| total                  | $3.069306930693069 \cdot 10^{-4}$ sec | 202   | 0.062 sec | 0      |


```