Notes on Sequences Generated by Repeated Application of Different Operators on Sequential Natural Numbers

Aaron Webster

October 12, 2011

1 Purpose

The purpose of this document is to give some convenient non-recursive ways of computing resultant sequences produced by the repeated application of operators on sequential natural numbers $\mathbb{N}$ (OEIS A000027). That is to say, for an operator $\diamond$ on a set of k numbers $n = \{n_1 \dots n_k : n_k \in \mathbb{N}\}$

$$n_1 = 0 \tag{1}$$
$$n_1 = 1 \tag{2}$$
$$n_2 = 2 \tag{3}$$
$$\dots \tag{4}$$

the resulting sequence a_n is defined by

$$a_k = 1 \diamond 2 \diamond 3 \diamond 4 \dots \diamond n_{k-1} \diamond n_k \tag{5}$$

In this document, n shall refer to OEIS A000027 and n_k the k^{th} term in that sequence.

2 Addition

The addition operator is represented by the + symbol. The sequence produced by addition

$$a_k = 1 + 2 + 3 + 4 \dots + n_{k-1} + n_k \tag{6}$$

is familiar in the notation of the sum

$$a_k = \sum_{i=0}^{k} n_i \tag{7}$$

The result is known as the *triangular numbers* (OEIS A000217). The first few terms in this sequence are

$$a_k = \{0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153, \dots\} \tag{8}$$

The formula for the k^{th} term of this sequence is given by

$$a_k = \frac{k(k+1)}{2} = \frac{k^2 + k}{2} \tag{9}$$

This is the first order case of *Faulhaber's formula*, used to compute certain types of polynomials of the same name. Note that since either k or $k + 1$ is even, $k(k + 1)/2$ is an integer and, with the exception of a_3, must be composite.

2.1 Subtraction

Since there are no negative numbers in $\mathbb{N}$, application of the subtraction operator is more or less nonsensical. Nonetheless it is trivial to see that the result of subtraction is identical to that of addition when the result is multiplied by -1.

3 Multiplication

The sequence produced by repeated applications of the multiplication operator $\times$ on n

$$a_k = 1 \times 2 \times 3 \times 4 \ldots \times n_{k-1} \times n_k \tag{10}$$

is also notationally familiar as the product

$$a_k = \prod_{i=1}^{k} n_i = k! \tag{11}$$

with the result being the factorial of k, and the sequence is known as the *factorial numbers* (OEIS A000142). Note that this sequence starts with n_1 instead of n_0. In this case, there is no analytic expression, but a number of interesting algorithms and their implementations may be found online at a page maintained by Peter Luschny [1]

4 Conjunction (AND)

The logical conjunction is represented by the operator $\wedge$. Its truth table is as follows.

p	q	$\wedge$
0	0	0
0	1	0
1	0	0
1	1	1

The result of repeated applications of the logical conjunction

$$a_k = 1 \wedge 2 \wedge 3 \wedge 4 \ldots \wedge n_{k-1} \wedge n_k \tag{12}$$

$$= \{0, 0, 0, 0, 0, 0, \ldots\} \tag{13}$$

is uninteresting. The sequence will be the same whether 1 or 0 is chosen as the starting value. It is mentioned here only for the sake of completeness.

5 Disjunction (OR)

The truth table of the logical disjunction, represented by the $\vee$ operator is

p	q	$\vee$
0	0	0
0	1	1
1	0	1
1	1	1

The sequence produced is OEIS A000122

$$a_k = 1 \vee 2 \vee 3 \vee 4 \ldots \vee n_{k-1} \vee n_k \tag{14}$$

$$= \{0, 1, 3, 3, 7, 7, 7, 7, 15, 15, 15, 15, 15, 15, 15, 15, 31, 31, 31, 31, \ldots\} \tag{15}$$

[1] `http://www.luschny.de/math/factorial/FastFactorialFunctions.htm`

Note that when a bit is set in the binary representation of the sequence a_k, it never becomes unset

$$a_0 = 0000b \tag{16}$$
$$a_1 = 0001b \tag{17}$$
$$a_2 = 0011b \tag{18}$$
$$a_3 = 0011b \tag{19}$$
$$a_4 = 0111b \tag{20}$$
$$a_5 = 0111b \tag{21}$$
$$a_6 = 0111b \tag{22}$$
$$a_7 = 0111b \tag{23}$$
$$a_8 = 1111b \tag{24}$$
$$\vdots \tag{25}$$

therefore a_k will always be one less than a power of two. The most intuitive way of accomplishing this is with the following C code, which is a modification of an algorithm for rounding a number up to the next highest power of two. [2]

```
a_k = k;
a_k |= a_k >> 1;
a_k |= a_k >> 2;
a_k |= a_k >> 4;
a_k |= a_k >> 8;
a_k |= a_k >> 16;
```

Given k, this returns a_k assuming k is a 32 bit integer. For a 64 bit integer, simply add the extra operation $k| = k >> 32$.

6 Exclusive Disjunction (XOR)

The exclusive disjunction is represented by the $\oplus$ operator. It's truth table is

p	q	$\oplus$
0	0	0
0	1	1
1	0	1
1	1	0

The resulting sequence is OEIS A003815

$$a_k = 1 \vee 2 \vee 3 \vee 4 \ldots \vee n_{k-1} \vee n_k \tag{26}$$
$$= \{0, 1, 3, 0, 4, 1, 7, 0, 8, 1, 11, 0, 12, 1, 15, 0, 16, 1, 19, 0, 20, 1, \ldots\} \tag{27}$$

To find a_k, the following piecewise definition may be used

$$m = k \bmod 4 \tag{28}$$
$$a_k = \begin{cases} k & \text{if } m = 0 \\ 1 & \text{if } m = 1 \\ k+1 & \text{if } m = 2 \\ 0 & \text{if } m = 3 \end{cases} \tag{29}$$

Alternatively, the following C code may be convenient as a way to avoid branch conditions.

[2] http://graphics.stanford.edu/~seander/bithacks.html#RoundUpPowerOf2

```
m = k % 4;
a_k = (m>>1)^(m&1)+(!(m&1))*k;
```

This essentially works by creating the bit pattern 0110 corresponding to the addition of one when $m = 1$ or $m = 2$ and adding it to the pattern 1010, corresponding to the addition of k to this result when $m = 0$ or $m = 2$.