Problem 19

```
Clear[wa, n];

wa[0] = 1; wa[1] = 2; wa[2] = 4; wa[3] = 8; wa[4] = 16;

wa[n_] := wa[n] = 2 * wa[n - 1] + wa[n - 5]

Table[wa[k], {k, 1, 10}]

{2, 4, 8, 16, 33, 68, 140, 288, 592, 1217}
```

Problem 22

```
{{1, 3}, {1, 2, 3}} (* n = 3 ; 2^(3-2) *)

{{1, 4}, {1, 2, 4}, {1, 3, 4}, {1, 2, 3, 4}} (* n = 4 ; 2^(4-2) *)

{{1, 5}, {1, 2, 5}, {1, 3, 5}, {1, 2, 3, 5},
 {1, 4, 5}, {1, 2, 4, 5}, {1, 3, 4, 5}, {1, 2, 3, 4, 5}}

{{1, 6}, {1, 2, 6}, {1, 3, 6}, {1, 2, 3, 6}, {1, 4, 6}, {1, 2, 4, 6},
 {1, 3, 4, 6}, {1, 2, 3, 4, 6}, {1, 5, 6}, {1, 2, 5, 6}, {1, 3, 5, 6},
 {1, 2, 3, 5, 6}, {1, 4, 5, 6}, {1, 2, 4, 5, 6}, {1, 3, 4, 5, 6}, {1, 2, 3, 4, 5, 6}}
```

This can be explained by bit strings of length $n-2$. The positions belong to the numbers 2, ..., $n-1$. 1-s will tell you which numbers to include in the sequence.

```
Clear[ns, n];

ns[2] = 1; ns[3] = 2; ns[n_] := ns[n] = 2 ns[n - 1]

Table[ns[k], {k, 2, 10}]

{1, 2, 4, 8, 16, 32, 64, 128, 256}
```

Problem 23

The problem is to find a recurrence realtion for the number of bit strings of length n which contain at least one occurence of the string 00.

Easily we calculate $a_0 = 0$, $a_1 = 0$, $a_2 = 1$, $a_3 = 3$. Let $n > 3$. The recursion is based on the position of the first occurrence of 00. There are $2\, a_{n-1}$ bit strings of length n in which the first occurence at the position k, $k \le n-2$. If the first occurence of 00 is at the position $n-1$, then the bit at the position $n-2$ is 1 and there are no occurences of 00 among the first $n-3$ bits. It has been calculated that there are Fibonacci[$n-1$] bit strings of length $n-3$ which do not have any occurence of 00. Thus the recursion is

```
Clear[sb]; sb[0] = 0; sb[1] = 0; sb[n_] := sb[n] = 2 * sb[n - 1] + Fibonacci[n - 1]
```

```
Table[sb[k], {k, 0, 10}]
```

```
{0, 0, 1, 3, 8, 19, 43, 94, 201, 423, 880}
```

```
Table[2^k - Fibonacci[k + 2], {k, 0, 10}]
```

```
{0, 0, 1, 3, 8, 19, 43, 94, 201, 423, 880}
```

```
Table[Length[Select[BiSt[n], MemberQ[Partition[#, 2, 1], {0, 0}] &]], {n, 2, 10}]
```

```
{1, 3, 8, 19, 43, 94, 201, 423, 880}
```

```
? Fib*
```

```
Fibonacci[n] gives the nth Fibonacci number. Fibonacci[
   n, x] gives the nth Fibonacci polynomial, using x as the variable. More…
```

```
Table[{Length[Select[BiSt[n], Not[MemberQ[Partition[#, 2, 1], {0, 0}]] &]],
  Fibonacci[n + 2]}, {n, 2, 10}]
```

```
{{3, 3}, {5, 5}, {8, 8}, {13, 13}, {21, 21}, {34, 34}, {55, 55}, {89, 89}, {144, 144}}
```

Easily we calculate $a_0 = 0$, $a_1 = 0$, $a_2 = 1$, $a_3 = 3$. Let $n > 3$. The recursion is based on the starting two bits. They can be 00, 01, or 1. There are three disjoint sets determined by these beginings. The cardinality of the first set is 2^{n-2}, the second set is a_{n-2} and the third set is a_{n-1}.

Problem 26

The problem is to find a recurrence realtion for the number of bit strings of length n which contain at least one occurence of the string 01.

Easily we calculate $a_0 = 0$, $a_1 = 0$, $a_2 = 1$, $a_3 = 4$. Let $n > 3$. The recursion is based on the position of the first occurrence of 01. There are $2\, a_{n-1}$ bit strings of length n in which the first occurence at the position k, $k \le n-2$. If the first occurence of 01 is at the last two bits, then the first $n-2$ positions do not include any 01-s. There are $n-1$ such bit strings: all 0-s, 1-s up to k, then zeros, $k = 1, \ ..., n-2$. Thus the recursion is

```
Clear[sa, n];
sa[0] = 0; sa[1] = 0; sa[n_] := sa[n] = 2 * sa[n - 1] + n - 1
```

```
Table[sa[k], {k, 0, 10}]
```

```
{0, 0, 1, 4, 11, 26, 57, 120, 247, 502, 1013}
```

```
Table[2^n - (n + 1), {n, 0, 10}]
```

```
{0, 0, 1, 4, 11, 26, 57, 120, 247, 502, 1013}
```

Or one can reason based on the ending bit. There are a_{n-1} strings ending with 0, there are 2^{n-2} strings ending with 01. How many strings end with 11? There are a_{n-2} strings ending with 11, but these are not all. There are $n - 2$ strings of the form 00...011, 100..011, and so on to, 11...1011. The listed sets are disjoint, so

```
Clear[sb, n];
sb[0] = 0; sb[1] = 0; sb[n_] := sb[n] = sb[n - 1] + 2^(n-2) + sb[n - 2] + n - 2
```

```
Table[sb[k], {k, 0, 10}]
```

```
{0, 0, 1, 4, 11, 26, 57, 120, 247, 502, 1013}
```

Or one can reason by building n-th from the previous. There are $2 * a_{n-1}$ bitstrings that have 01 in the first n bits. There are $2^{n-2} - a_{n-2}$ bitstrings with no 01 in the first n-2 bits. End them with 01. These sets are disjoint.

```
Clear[sc, n];
sc[0] = 0; sc[1] = 0; sc[n_] := sc[n] = 2 * sc[n - 1] + 2^(n-2) - sc[n - 2]
```

```
Table[sc[k], {k, 0, 10}]
```

```
{0, 0, 1, 4, 11, 26, 57, 120, 247, 502, 1013}
```

■ Testing

```
bits8 = PadLeft[#, 8] & /@ Table[IntegerDigits[k, 2], {k, 0, 2^8 - 1}];
```

```
Length[bits8]
```

```
256
```

```
MemberQ[Partition[bits8[[5]], 2, 1], {0, 1}]
```

```
True
```

```
BiSt[n_] := PadLeft[#, n] & /@ Table[IntegerDigits[k, 2], {k, 0, 2^n - 1}]
```

```
Length[Select[BiSt[4], MemberQ[Partition[#, 2, 1], {0, 1}] &]]
```

```
11
```

```
Length[Select[BiSt[8], MemberQ[Partition[#, 2, 1], {0, 1}] &]]
```

```
247
```

```
Length[Select[BiSt[10], MemberQ[Partition[#, 2, 1], {0, 1}] &]]
```

```
1013
```

```
Select[BiSt[4], MemberQ[Partition[#, 2, 1], {0, 1}] &]
```

```
{{0, 0, 0, 1}, {0, 0, 1, 0}, {0, 0, 1, 1}, {0, 1, 0, 0}, {0, 1, 0, 1},
 {0, 1, 1, 0}, {0, 1, 1, 1}, {1, 0, 0, 1}, {1, 0, 1, 0}, {1, 0, 1, 1}, {1, 1, 0, 1}}
```

```
Table[Length[Select[BiSt[n], MemberQ[Partition[#, 2, 1], {0, 1}] &]], {n, 2, 10}]
```

```
{1, 4, 11, 26, 57, 120, 247, 502, 1013}
```

27

Count the number of ways to climb n stairs if we can take either 1 or 2 stairs at the time.

For example 3 stairs:

```
{{1, 2}, {2, 1}, {1, 1, 1}}
```

or 4 stairs

```
{{1, 1, 2}, {2, 2}, {1, 2, 1}, {2, 1, 1}, {1, 1, 1, 1}}
```

```
Clear[st]; st[1] = 1; st[2] = 2; st[n_] := st[n] = st[n - 1] + st[n - 2]
```

```
Table[st[k], {k, 1, 20}]
```

```
{1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946}
```

29

a0 = 1; a1 = 3; a2 = 8;

```
{{}, {0}, {1}, {2}}
```

```
Length[{{1, 0}, {2, 0}, {0, 1}, {0, 2}, {1, 1}, {1, 2}, {2, 1}, {2, 2}}]
```

```
8
```

Split the set of all ternary strings of length n with no consecutive 0s into disjoint subsets: beginning with 1, begining with 2 and beginning with 0. How many of each?

```
Clear[ts]; ts[0] = 1; ts[1] = 3; ts[n_] := ts[n] = 2 * ts[n - 1] + 2 * ts[n - 2]
```

```
Table[ts[k], {k, 1, 10}]
```

```
{3, 8, 22, 60, 164, 448, 1224, 3344, 9136, 24960}
```