Bell numbers: how many ways to partition a set with n elements?

Start with the only possible partitioning of a 1 element set

a	1

From partitions on a N element set think of all the ways to add another element.

red: without changing the number of partitions, the new value must join an existing partition

blue: the only way to change the number of partitions is by adding the new item as its own partition.

a	ab	1
	a|b	1

one one times one	ab	abc	1
		ab|c	2
two one times one	a|b	ac|b	2
		a|bc	2
		a|b|c	3

	one one times one	abc	abcd	1
			abc|d	2
	two one times three	ab|c	abd|c	2
			ab|dc	2
			ab|c|d	3
		ac|b	acd|b	2
			ac|bd	2
			ac|b|d	3
		a|bc	ad|bc	2
			a|bcd	2
			a|bc|d	3
	three one times one	a|b|c	ad|b|c	3
			a|bd|c	3
			a|b|cd	3
			a|b|c|d	4

When moving from a set of size S to a set of size S+1,
If a particular partition has P subsets
then it will result in P (red) partitions, also of P subsets for the new set,
and one (blue) partition with P+1 subsets.
A set of size S will have at least one partition with every number from 1 to S of subsets

		one one times one	abcd	abcde	1
				abcd|e	2
		two one times seven	abc|d	abce|d	2
				abc|de	2
				abc|d|e	3
			abd|c	abde|c	2
				abc|ce	2
				abd|c|e	3
			ab|dc	abe|cd	2
				ab|cde	2
				ab|dc|e	3
			acd|b	acde|b	2
				acd|be	2
				acd|b|e	3
			ac|bd	ace|bd	2
				ac|bde	2
				ac|bd|e	3
			ad|bc	ade|bc	2
				ad|bce	2
				ad|bc|e	3
			a|bcd	ae|bcd	2
				a|bcde	2
				a|bcd|e	3
		three one times six	ab|c|d	abe|c|d	3
				ab|ce|d	3
				ab|c|de	3
				ab|c|d|e	4
			ac|b|d	ace|b|d	3
				ac|be|d	3
				ac|b|de	3
				ac|b|d|e	4
			ad|b|c	ade|b|c	3
				ad|be|c	3
				ad|b|ce	3
				ad|b|c|e	4
			a|bc|d	ae|bc|d	3
				a|bce|d	3
				a|bc|de	3
				ad|b|c|e	4
			a|bd|c	ae|bd|c	3
				a|bde|c	3
				a|bd|ce	3
				a|bd|c|e	4
			a|b|cd	ae|b|cd	3
				a|be|cd	3
				a|b|cde	3
				a|b|cd|e	4
		four one times one	a|b|c|d	ae|b|c|d	4
				a|be|c|d	4
				a|b|ce|d	4
				a|b|c|de	4
				a|b|c|d|e	1

The Bell number B(n) is the number of partitions a set of size n has.
It is easy to compute from a different starting point.

NP(S, P) is the number of partitions containing exactly P subsets that a set of size S would have.

From the tables, just counting:

NP(1, 1) = 1
NP(2, 1) = 1, NP(2, 2) = 1
NP(3, 1) = 1, NP(3, 2) = 3, NP(3, 3) = 1
NP(4, 1) = 1, NP(4, 2) = 7, NP(4, 3) = 6, NP(4, 4) = 1
NP(5, 1) = 1, NP(5, 2) = 15 , NP(5, 3) = 25, NP(5, 4) = 10, NP(5, 5) = 1

All of that in the expected dynamic programming form:

S P	1	2	3	4	5
1	1				
2	1	1			
3	1	3	1		
4	1	7	6	1	
5	1	15	25	10	1

Obviously

NP(anything, 1) = 1 (only 1 subset, must hold everything), and
NP(anything, the same thing) = 1 (S singleton subsets), and
NP(anything, anything bigger) = 0

S P	1	2	3	4	5
1	1	0	0	0	0
2	1	1	0	0	0
3	1		1	0	0
4	1			1	0
5	1				1

The note was that a partition of a set of size S that has P subsets contributes:
P partitions, also of size P, and
1 partition of size P+1
to the partitions of the new set of size S+1

So NP(S+1, P) = P * NP(S, P) + NP(S, P−1)
P for each of the same size plus
1 for each of the smaller size

P * NP(S, P) is column * above
NP(S, P-1) is left-and-above

So the next iteration is:

S P	1	2	3	4	5
1	1	0	0	0	0
2	1	1	0	0	0
3	1	2 + 1 = 3	1	0	0
4	1	6 + 1 = 7		1	0
5	1	14 + 1 = 15			1

And the next:

S P	1	2	3	4	5
1	1	0	0	0	0
2	1	1	0	0	0
3	1	3	1	0	0
4	1	7	3 + 3 = 6	1	0
5	1	15	18 + 7 = 25		1

And the next:

S P	1	2	3	4	5
1	1	0	0	0	0
2	1	1	0	0	0
3	1	3	1	0	0
4	1	7	6	1	0
5	1	15	25	4 + 6 = 10	1

It works!

And finally B(n) = sum of all NP(n, anything)

B(1) = 1, B(2) = 2, B(3) = 5, B(4) = 15, B(5) = 52.

I am certain there is no (known) closed formula.