Section 4.2 — Example 9 Teacher Manual
Counting Bit Additions in Algorithm 2

Detailed explanation (matching Example 9)

At each bit position j Algorithm 2 computes

$$d = \left\lfloor \frac{a_j+b_j+c}{2} \right\rfloor, \qquad s_j = a_j + b_j + c - 2d, \qquad c \leftarrow d.$$

Implementing the three-bit sum $a_j + b_j + c$ uses at most two one-bit additions:

$$t_1 = (a_j + b_j) \quad \text{and} \quad t_2 = t_1 + c.$$

If $c = 0$ and $a_j + b_j < 2$, the second add is effectively a no-op, so the total count is *strictly less than* $2n$. In any case, the bound $\leq 2n$ yields linear time $O(n)$.

Worked examples

A. Two-bit example: $(11)_2 + (01)_2$

j	a_j	b_j	c (in)	s_j	c (out)
0	1	1	0	0	1
1	1	0	1	0	1

$s_2 = 1$

Result: $(100)_2$. Bit additions used ≤ 4.

B. Four-bit example: $(1011)_2 + (0110)_2$

j	a_j	b_j	c (in)	s_j	c (out)
0	1	0	0	1	0
1	1	1	0	0	1
2	0	1	1	0	1
3	1	0	1	0	1

$s_4 = 1$

Result: $(10001)_2$. Bit additions used ≤ 8.

Solutions to student practice

1) $(0101)_2 + (0011)_2$

j	a_j	b_j	c (in)	s_j	c (out)
0	1	1	0	0	1
1	0	1	1	0	1
2	1	0	1	0	1
3	0	0	1	1	0

$s_4 = 0$

Answer: $(10000)_2$.

2) $(1001)_2 + (0001)_2$

j	a_j	b_j	c (in)	s_j	c (out)
0	1	1	0	0	1
1	0	0	1	1	0
2	0	0	0	0	0
3	1	0	0	1	0

$s_4 = 0$

Answer: $(1010)_2$.

3) $(11101101)_2 + (10111011)_2$

Work right-to-left; lots of carries chain through:

j	a_j	b_j	c (in)	s_j	c (out)
0	1	1	0	0	1
1	0	1	1	0	1
2	1	1	1	1	1
3	1	0	1	0	1
4	0	1	1	0	1
5	1	1	1	1	1
6	1	0	1	0	1
7	1	1	1	1	1

$s_8 = 1$

Answer: $(1\,011010000)_2 = (1011010000)_2$.

Bit-addition count bound. In all three problems, the number of one-bit additions is $< 2n$ and $\leq 2n$, so linear in the input length.