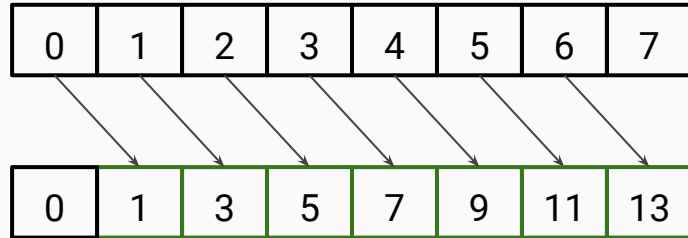
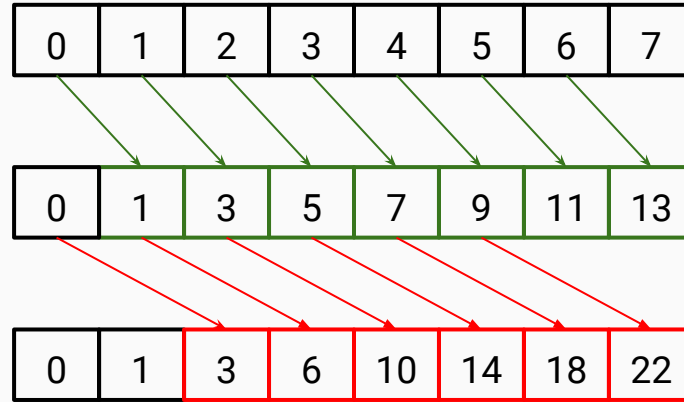


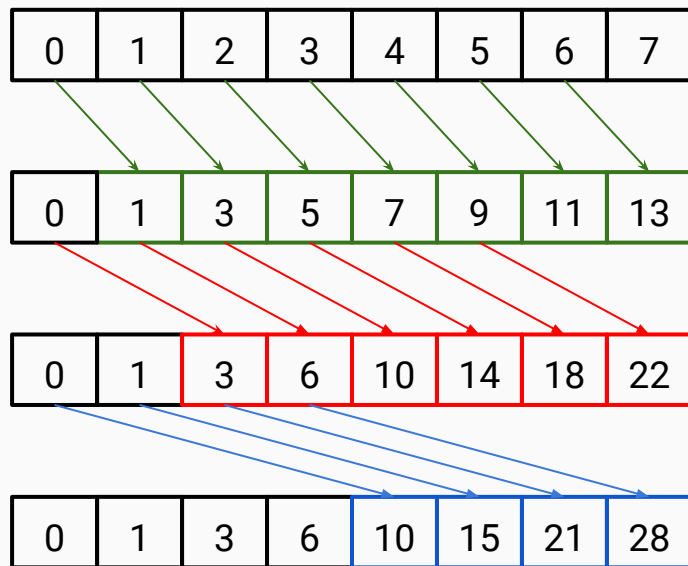
În acest exemplu, fiecare proces are câte un element (8 elemente - 8 procese)



- procesul 0 trimite procesului 1, care adună valoarea primită ( $0 + 1 = 1$ )
- procesul 1 trimite procesului 2, care adună valoarea primită ( $1 + 2 = 3$ )
- procesul 2 trimite procesului 3, care adună valoarea primită ( $2 + 3 = 5$ )
- procesul 3 trimite procesului 4, care adună valoarea primită ( $3 + 4 = 7$ )
- procesul 4 trimite procesului 5, care adună valoarea primită ( $4 + 5 = 9$ )
- procesul 5 trimite procesului 6, care adună valoarea primită ( $5 + 6 = 11$ )
- procesul 6 trimite procesului 7, care adună valoarea primită ( $6 + 7 = 13$ )



- procesul 0 trimite procesului 2, care adună valoarea primită ( $0 + 3 = 3$ )
- procesul 1 trimite procesului 3, care adună valoarea primită ( $1 + 5 = 6$ )
- procesul 2 trimite procesului 4, care adună valoarea primită ( $3 + 7 = 10$ )
- procesul 3 trimite procesului 5, care adună valoarea primită ( $5 + 9 = 14$ )
- procesul 4 trimite procesului 6, care adună valoarea primită ( $7 + 11 = 18$ )
- procesul 5 trimite procesului 7, care adună valoarea primită ( $9 + 13 = 22$ )



- procesul 0 trimite procesului 4, care adună valoarea primită ( $0 + 10 = 10$ )
- procesul 1 trimite procesului 5, care adună valoarea primită ( $1 + 14 = 15$ )
- procesul 2 trimite procesului 6, care adună valoarea primită ( $3 + 18 = 21$ )
- procesul 3 trimite procesului 7, care adună valoarea primită ( $6 + 22 = 28$ )

```
for (pas = 1; pas < nr_procese; pas *= 2)
    if (rank + pas < nr_procese)
        trimite la procesul cu rank-ul [rank + pas]
    if (rank - pas >= 0)
        primește de la procesul cu rank-ul [rank - pas]
    adună
```

pas = 1:

- 0 trimite lui 1 ( $0 + 1$ )
- 1 primește de la 0 ( $1 - 1$ )
  
- 1 trimite lui 2 ( $1 + 1$ )
- 2 primește de la 1 ( $2 - 1$ )
  
- 2 trimite lui 3 ( $2 + 1$ )
- 3 primește de la 2 ( $3 - 1$ )
  
- 3 trimite lui 4 ( $3 + 1$ )
- 4 primește de la 3 ( $4 - 1$ )

pas = 1:

- 4 trimite lui 5 ( $4 + 1$ )
- 5 primește de la 4 ( $5 - 1$ )
  
- 5 trimite lui 6 ( $5 + 1$ )
- 6 primește de la 5 ( $6 - 1$ )
  
- 6 trimite lui 7 ( $6 + 1$ )
- 7 primește de la 6 ( $7 - 1$ )

## Scan - pseudocod - rulare

pas = 2:

- 0 trimite lui 2 ( $0 + 2$ )
- 2 primește de la 0 ( $2 - 2$ )
  
- 1 trimite lui 3 ( $1 + 2$ )
- 3 primește de la 1 ( $3 - 2$ )
  
- 2 trimite lui 4 ( $2 + 2$ )
- 4 primește de la 2 ( $4 - 2$ )
  
- 3 trimite lui 5 ( $3 + 2$ )
- 5 primește de la 3 ( $5 - 2$ )
  
- 4 trimite lui 6 ( $4 + 2$ )
- 6 primește de la 4 ( $6 - 2$ )
  
- 5 trimite lui 7 ( $5 + 2$ )
- 7 primește de la 5 ( $7 - 2$ )

pas = 4:

- 0 trimite lui 4 ( $4 + 4$ )
- 4 primește de la 0 ( $4 - 4$ )
  
- 1 trimite lui 5 ( $1 + 4$ )
- 5 primește de la 1 ( $5 - 1$ )
  
- 2 trimite lui 6 ( $2 + 4$ )
- 6 primește de la 2 ( $6 - 4$ )
  
- 3 trimite lui 7 ( $3 + 4$ )
- 7 primește de la 3 ( $7 - 4$ )