Compressed Sparse Row (CSR) Format

$$\begin{bmatrix} 1 & 0 & 0 & 3 & 5 \\ 0 & 3 & 0 & 0 & 8 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

COO-Format

nz_id	0	1	2	3	4	....	
row	0	0	0	1	1	....	→ n_nz int
row_ptr.	0	–	–	3	–	....	→ (n_rows+1) int
col	0	3	4	1	4	....	→ n_nz int
val	1	3	5	3	8	....	→ n_nz double

repeated !

→ CSR: save all col_ids, only save row ptrs
save all values

global variables

- → n_rows (1 int)
- → n_cols (1 int)
- → n_nz (1 int)

local variables

- → row_ptrs (n_rows +1 int)
- → col_ids (n_nz int)
- → values (n_nz double)

if sizeof(int) = sizeof(double) = 8 Byte

total size = (3 + n_rows + 1 + 2·n_nz) · 8 Byte

Example

$$\underline{\underline{A}} = \begin{bmatrix} 2 & 0 & 0 & 2 & 0 \\ 3 & 4 & 2 & 5 & 0 \\ 5 & 0 & 0 & 8 & 17 \\ 0 & 0 & 10 & 16 & 0 \\ 0 & 0 & 0 & 0 & 14 \end{bmatrix}$$

5 rows

5 cols

n_nz = 12

nz_id	0	1	2	3	4	5	6	7	8	9	10	11		
row_ptr	0	–	2	–	–	–	6	–	–	9	–	11	12	IP
col_id	0	3	0	1	2	3	0	3	4	2	3	4		J
val	2	2	3	4	2	5	5	8	17	10	16	14		V

0-th row — 1st row — ...

Matrix-Vector Product : $\underline{\underline{A}}\,\underline{b} = \underline{c}$ (given, given, want)

```
for i=0:n_rows
    c[i]=0
end
for i=0:n_rows
    nz_start = IP[i]
    nz_end   = IP[i+1]
    for nz_id = nz_start : nz_end
        j = J[nz_id]
        val = V[nz_id]
        c[i] = c[i] + val · b[j]
    end
end
```

that's why we need [n_rows+1]