Pairwise alignment

Sequences

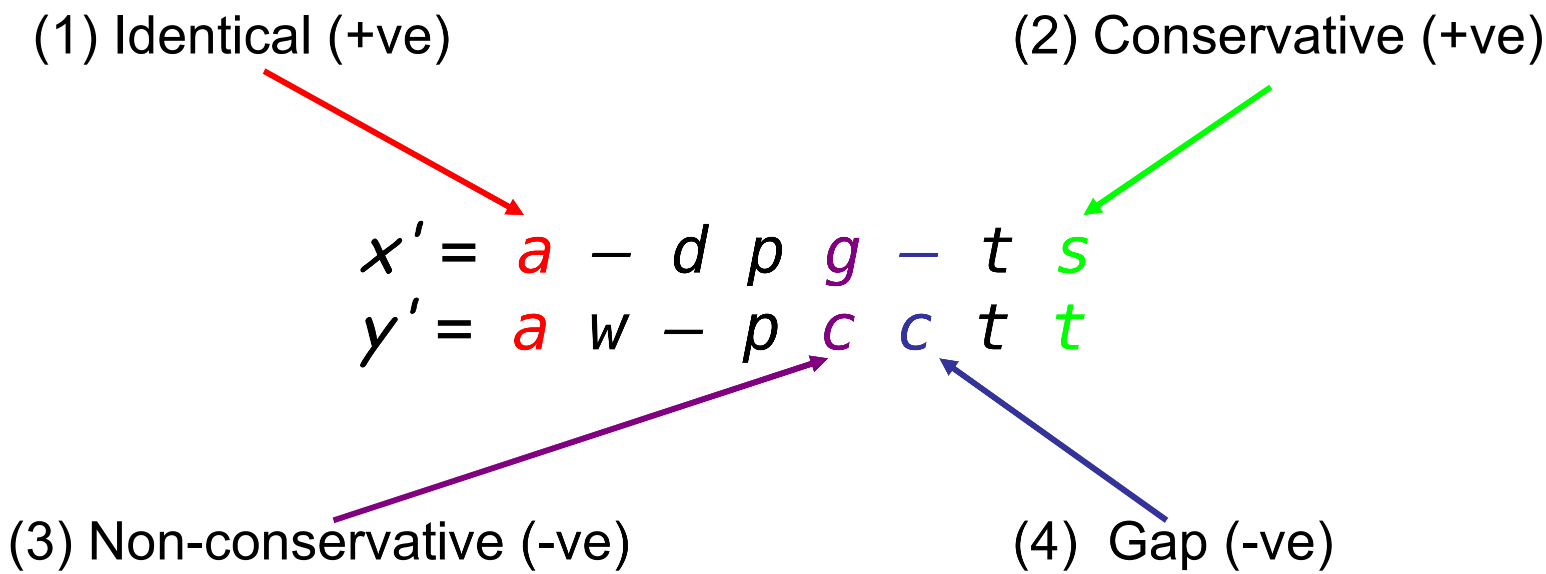
$$x = a \quad d \quad p \quad g \quad t \quad s$$
$$y = a \quad w \quad p \quad c \quad c \quad t \quad t$$

Alignment

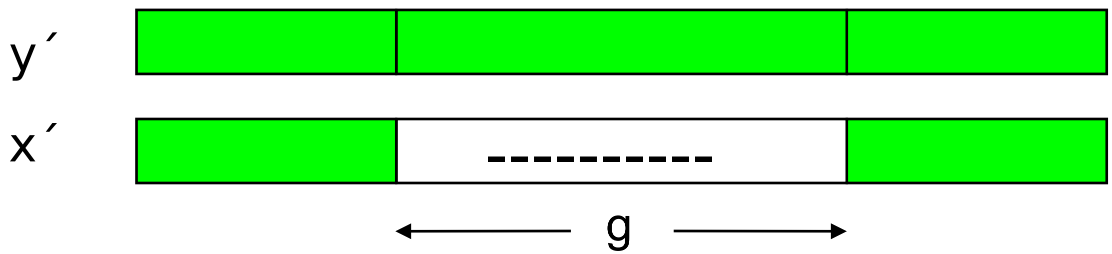
$$x' = a \quad - \quad d \quad p \quad g \quad - \quad t \quad s$$
$$y' = a \quad w \quad - \quad g \quad c \quad c \quad t \quad t$$

Scoring

- Numeric score associated with each column
- Total score = sum of column scores
- Column types:



Gap penalties



- **Linear** score: $\gamma\ (g) = -g\mathbf{d}$

gap penalty

- **Affine** score: $\gamma\ (g) = -\mathbf{d} - (g-1)\mathbf{e}$

gap-open penalty *gap-extension penalty*

How many alignments?

Depends on what we mean by “Alignments A and B are different. ”

For example,

```
AC-        A-C
A-T        AT-
```

If we agree they are not distinct, the number of possible alignments g(n,m) of two sequences of length n and m is $\binom{m+n}{n}$

$$g(n,n) = \binom{2n}{n} \approx \frac{4^n}{\sqrt{\pi n}}$$

$$g(21,21) \approx 10^{27}$$

Needleman & Wunsch algorithm

- Dynamic programming algorithm for global alignment
- Needleman & Wunsch (‘70), modified Gotoh (‘82)

Assumptions:

Linear gap score *d*

Symmetric scoring matrix *S*

$s(a,b) = s(b,a)$ score from lining up *a* and *b*

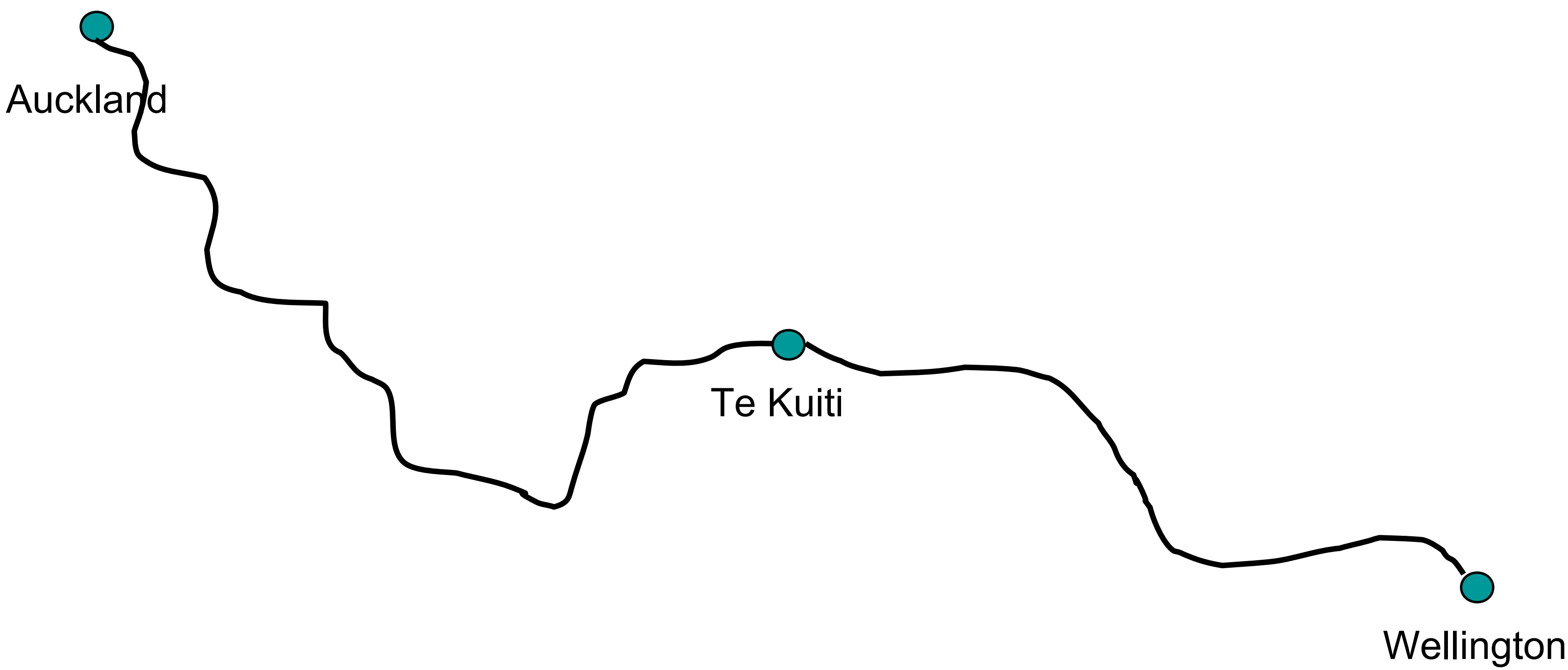
$s(a,-) = s(-,a) = -d$ score from lining up *a* with -

Dynamic Programming

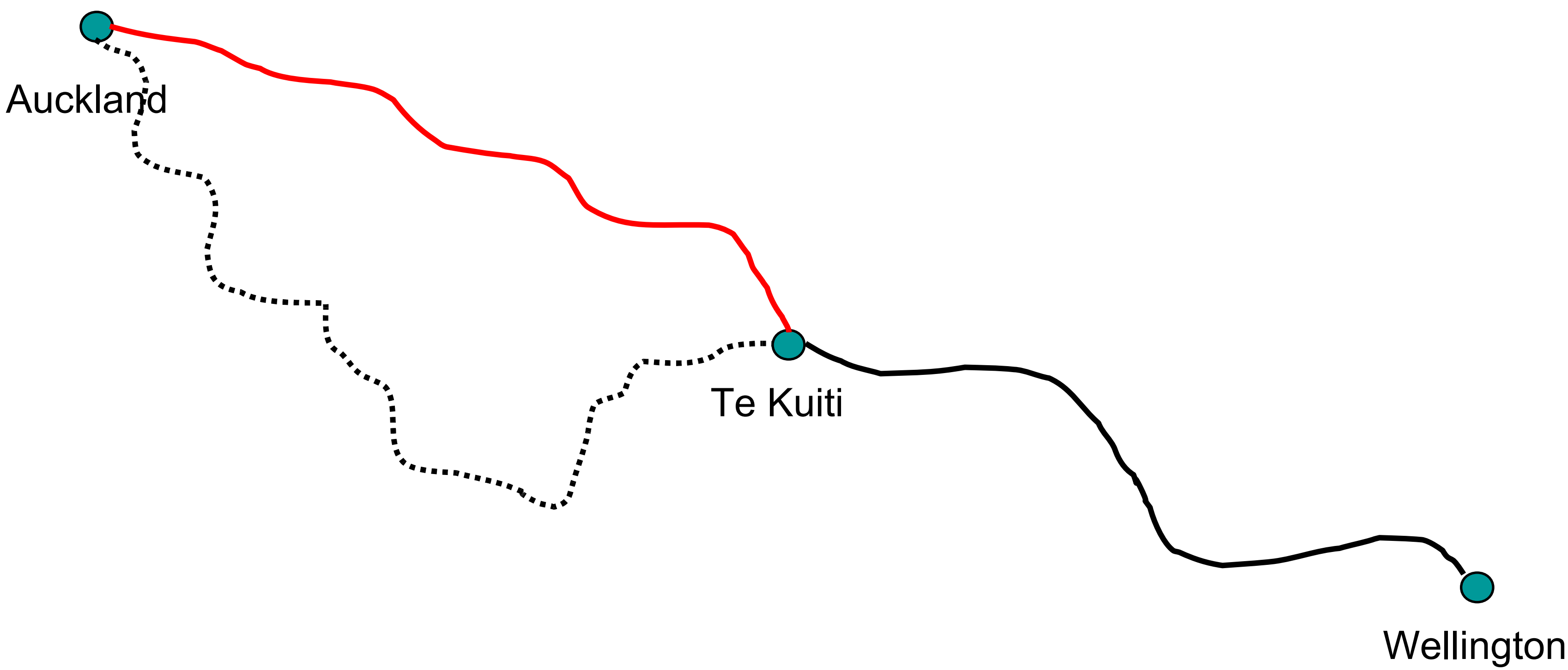
- method for solving combinatorial optimization problems
- guaranteed to give optimal solution
- generalization of "divide-and-conquer"
- relies on "Principle of Optimality"

  i.e. sub-optimal solution of sub-problem cannot be part of optimal solution of original problem instance.

Principle of Optimality



Principle of Optimality



Principle of Optimality

Given sequences:

$$Y = (y_1, y_2, \ldots, y_n)$$
$$X = (x_1, x_2, \ldots, x_m)$$

Define:

F(i,j) = score of best alignment

between $(y_1, y_2, \ldots, y_j)$

and $(x_1, x_2, \ldots, x_i)$

Principle of Optimality

Optimal alignment $\left\{ \begin{array}{l} y_1, y_2, y_3, \ldots, y_j \\ x_1, x_2, x_3, \ldots, x_i \end{array} \right.$ $F(i, j)$

Principle of Optimality

Optimal alignment

$$\begin{array}{l} y_1,\ y_2,\ y_3,\ \ldots,\ y_j \\ x_1,\ x_2,\ x_3,\ \ldots,\ x_i \end{array} \qquad F(i,j)$$

Looks like ……

$$\begin{array}{l|l} y_1, y_2, y_3, \ldots, y_{j-1} & y_j \\ x_1, x_2, x_3, \ldots, x_{i-1} & x_i \end{array} \qquad F(i-1, j-1) + s(x_i, y_j)$$

Principle of Optimality

Optimal alignment

$$\begin{array}{l} y_1, y_2, y_3, \ldots, y_j \\ x_1, x_2, x_3, \ldots, x_i \end{array} \qquad F(i, j)$$

Looks like ……

$$\begin{array}{l|l} y_1, y_2, y_3, \ldots, y_{j-1} & y_j \\ x_1, x_2, x_3, \ldots, x_{i-1} & x_i \end{array} \qquad F(i-1, j-1) + s(x_i, y_j)$$

or ……………

$$\begin{array}{l|l} y_1, y_2, y_3, \ldots, y_{j-1} & y_j \\ x_1, x_2, x_3, \ldots, x_i & - \end{array} \qquad F(i, j-1) - d$$

Principle of Optimality

Optimal alignment	$y_1, y_2, y_3, \ldots, y_j$ $x_1, x_2, x_3, \ldots, x_i$	$F(i,j)$
Looks like ……	$y_1, y_2, y_3, \ldots, y_{j-1}$ | y_j $x_1, x_2, x_3, \ldots, x_{i-1}$ | x_i	$F(i-1, j-1) + s(x_i, y_j)$
or ……………	$y_1, y_2, y_3, \ldots, y_{j-1}$ | y_j $x_1, x_2, x_3, \ldots, x_i$ | $-$	$F(i, j-1) - d$
or ……………	$y_1, y_2, y_3, \ldots, y_j$ | $-$ $x_1, x_2, x_3, \ldots, x_{i-1}$ | x_i	$F(i-1, j) - d$

Principle of Optimality

Optimal alignment

$$\begin{cases} y_1, y_2, y_3, \dots, y_j \\ x_1, x_2, x_3, \dots, x_i \end{cases} \qquad F(i,j)$$

Looks like ……

$$\begin{cases} y_1, y_2, y_3, \dots, y_{j-1} \;\; y_j \\ x_1, x_2, x_3, \dots, x_{i-1} \;\; x_i \end{cases} \qquad F(i-1, j-1) + s(x_i, y_j)$$

or ……………

$$\begin{cases} y_1, y_2, y_3, \dots, y_{j-1} \;\; y_j \\ x_1, x_2, x_3, \dots, x_i \;\; - \end{cases} \qquad F(i, j-1) - d$$

or ……………

$$\begin{cases} y_1, y_2, y_3, \dots, y_j \;\; - \\ x_1, x_2, x_3, \dots, x_{i-1} \;\; x_i \end{cases} \qquad F(i-1, j) - d$$

so ……………

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i, j-1) - d \\ F(i-1, j) - d \end{cases}$$

Principle of Optimality

Basis:

$$F(0,0) = 0$$

$$F(i,0) = F(i-1,0) + s(x_i, \text{-})$$

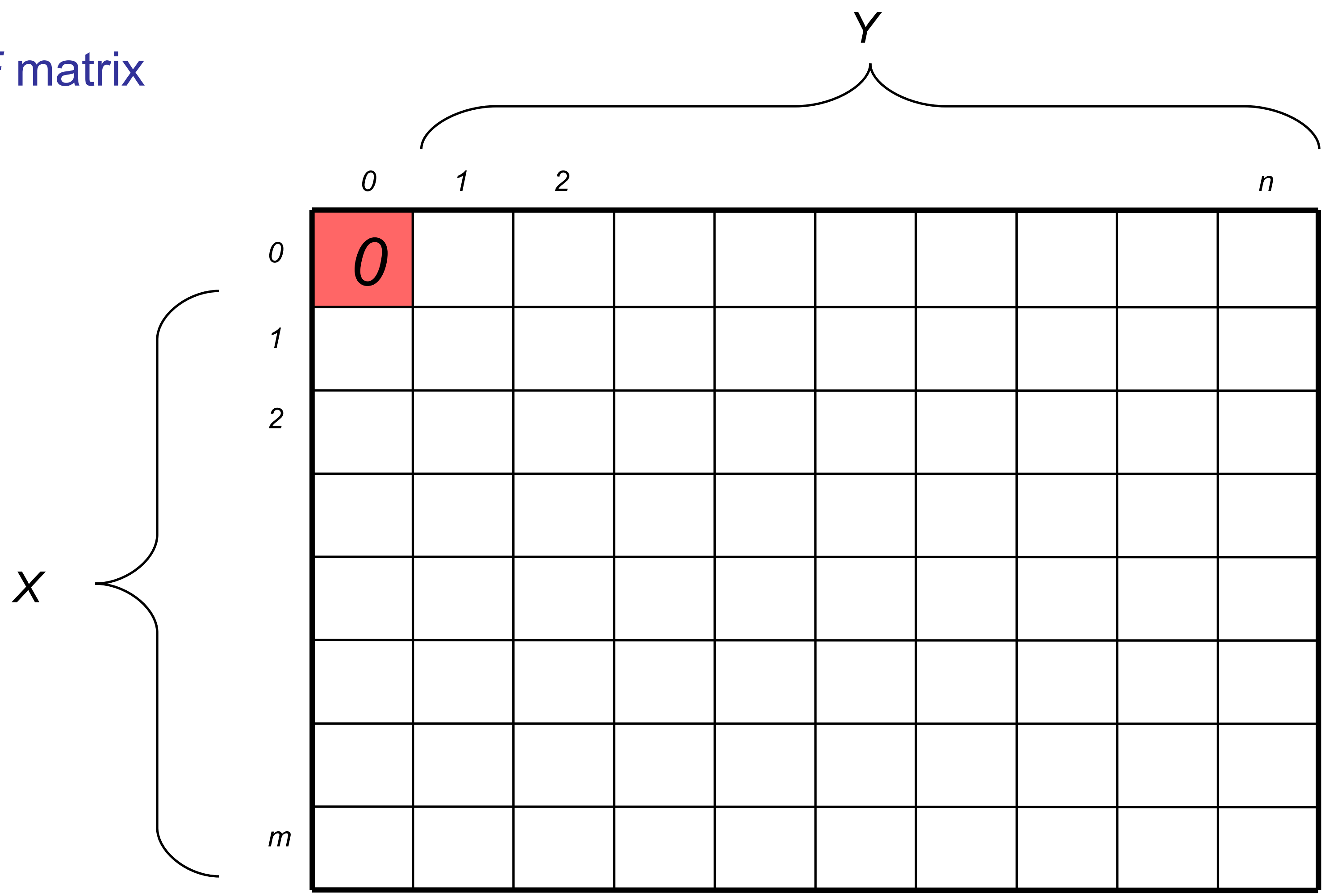
—	—	—	—	…	—
x_1,	x_2,	x_3,		…,	x_i

$$F(0,j) = F(0,j-1) + s(\text{-}, y_j)$$

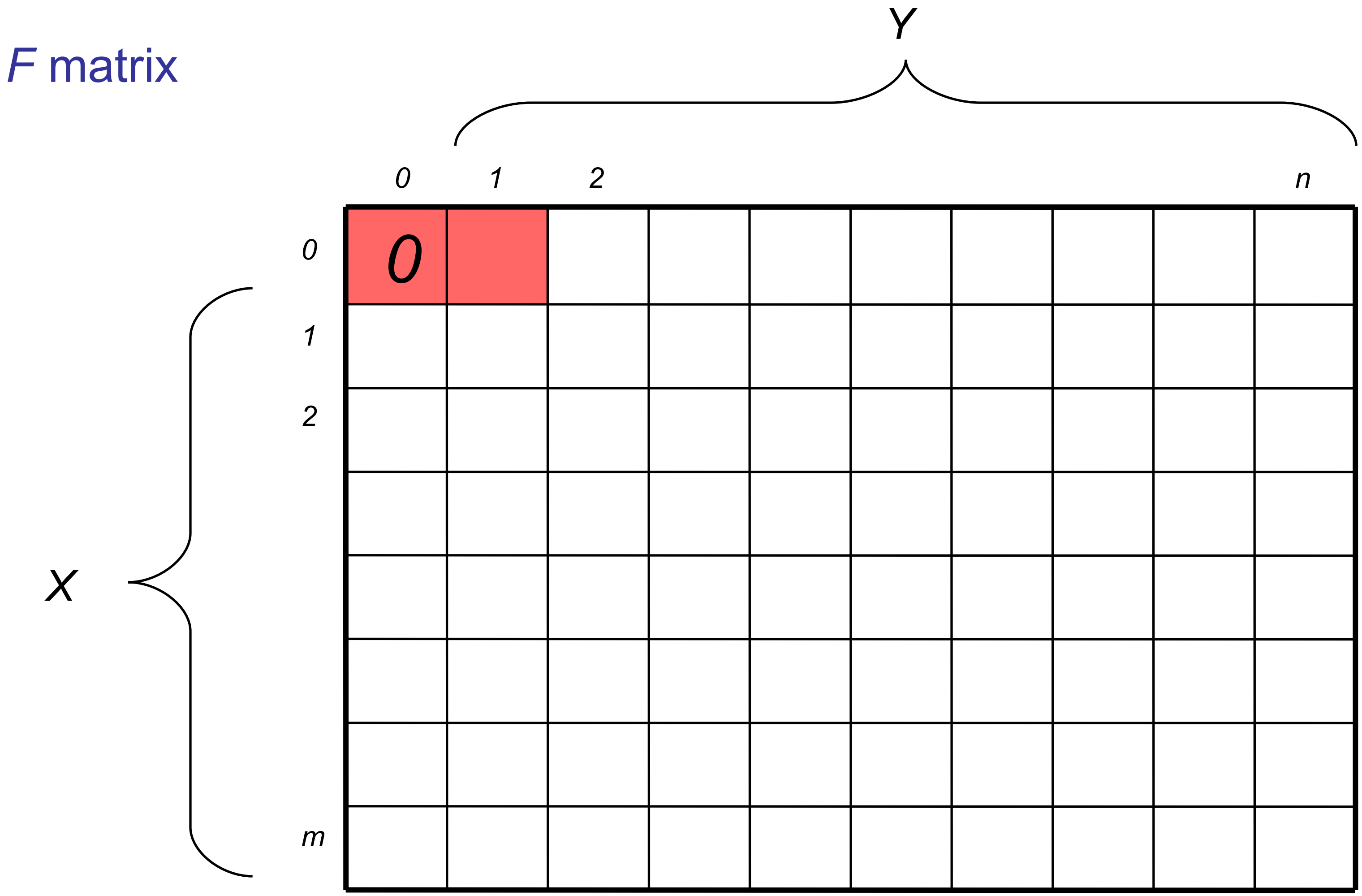
y_1,	y_2,	y_3,		…,	y_j
—	—	—	—	…	—

Filling up table

F matrix

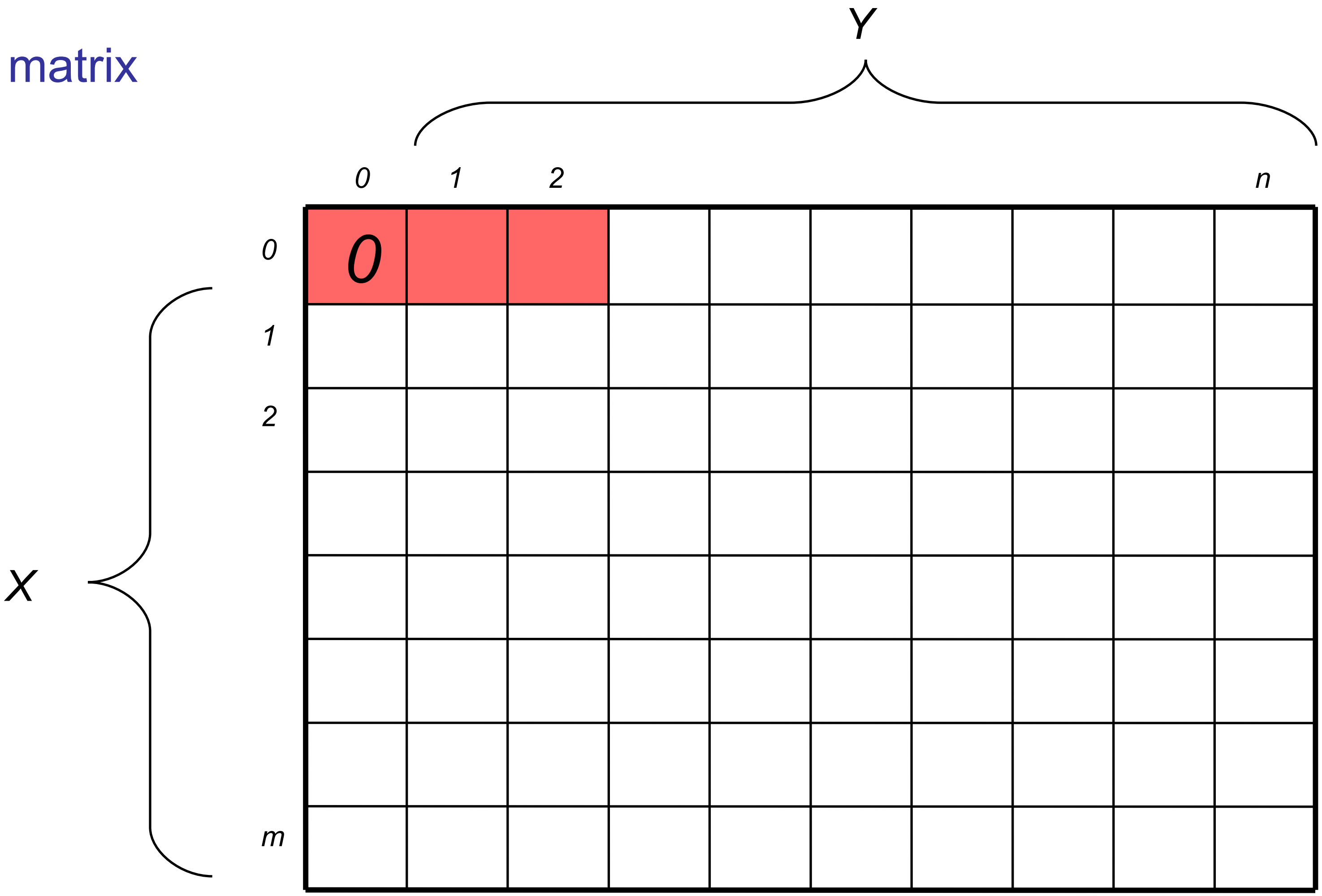


Filling up table



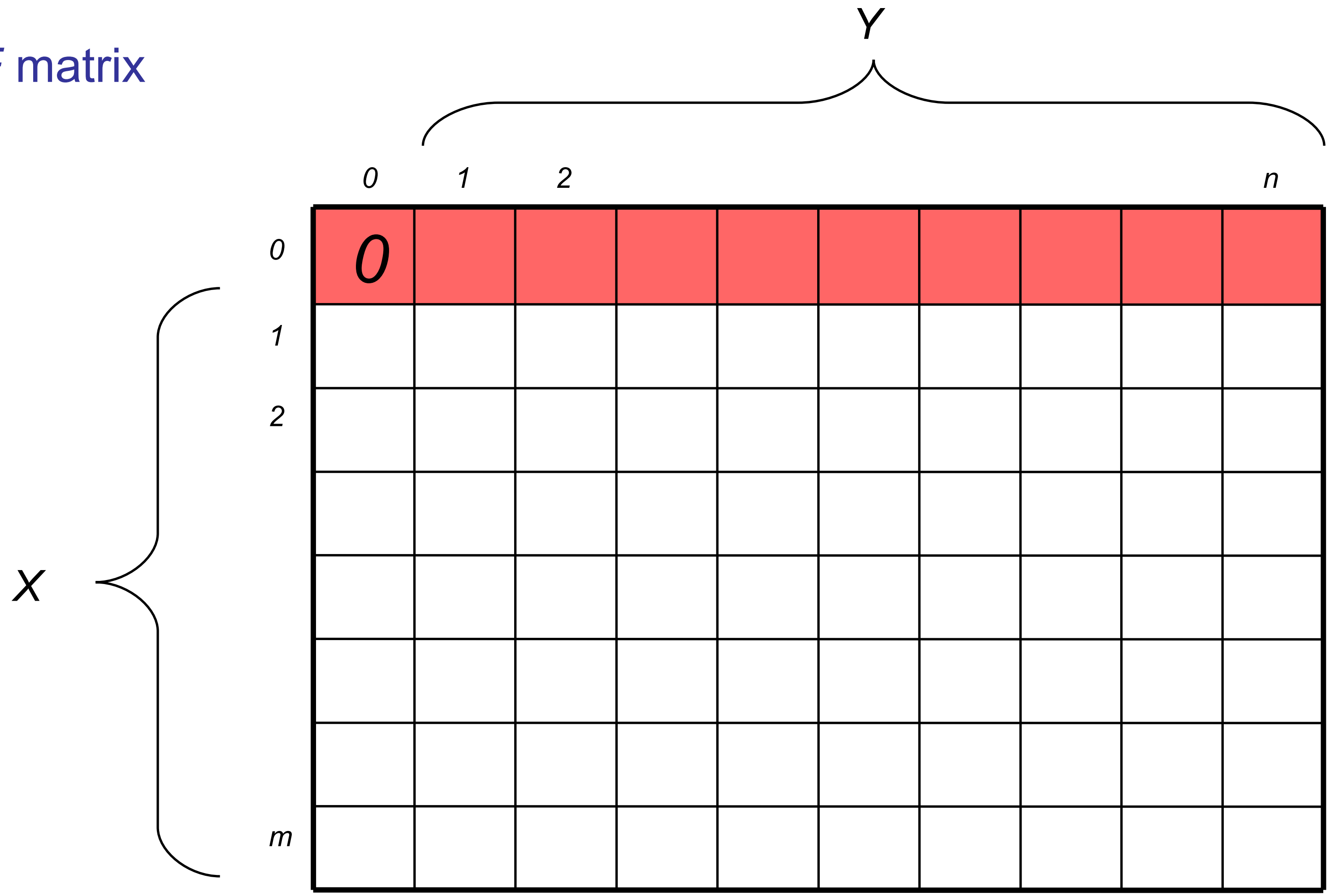
Filling up table

F matrix



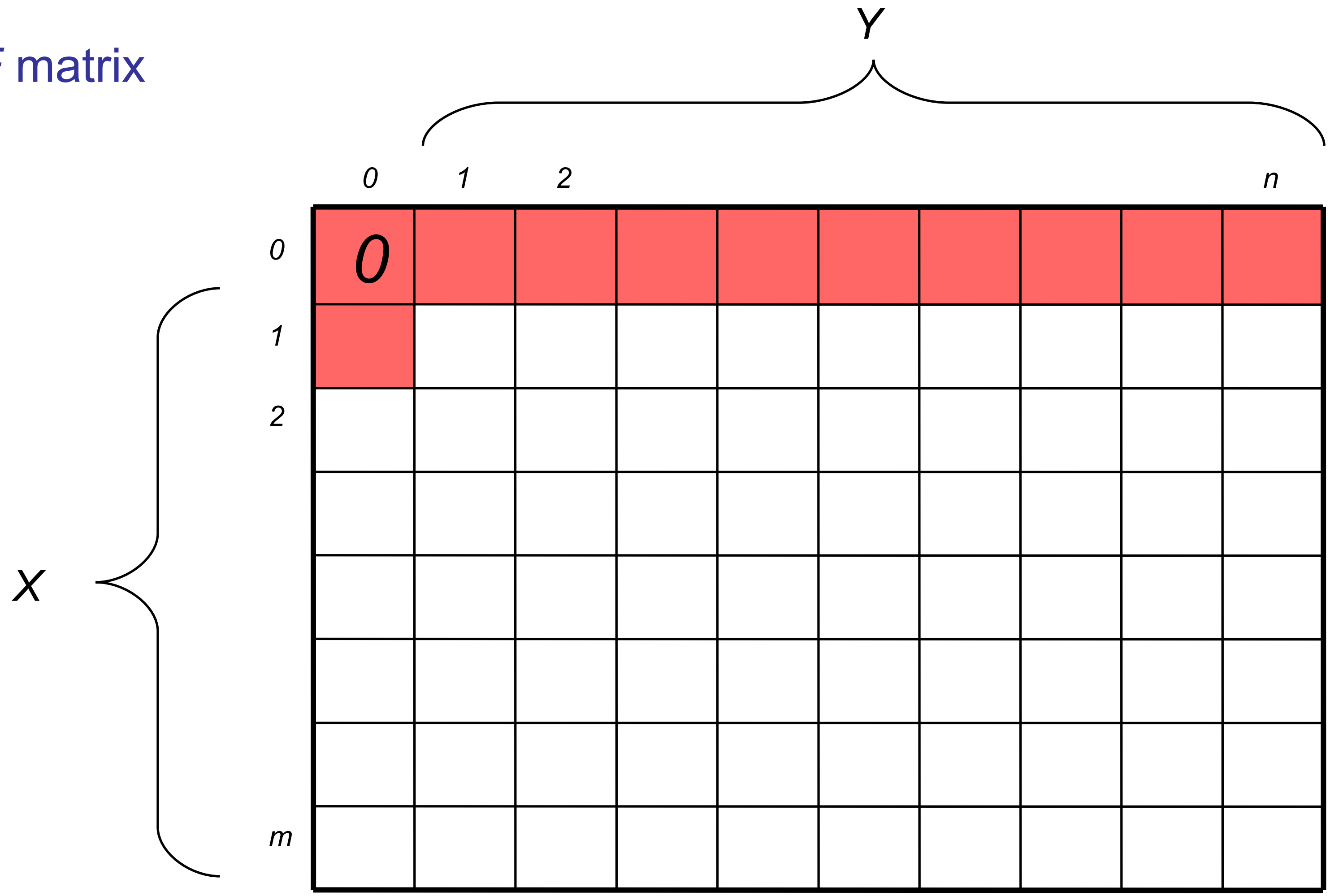
Filling up table

F matrix



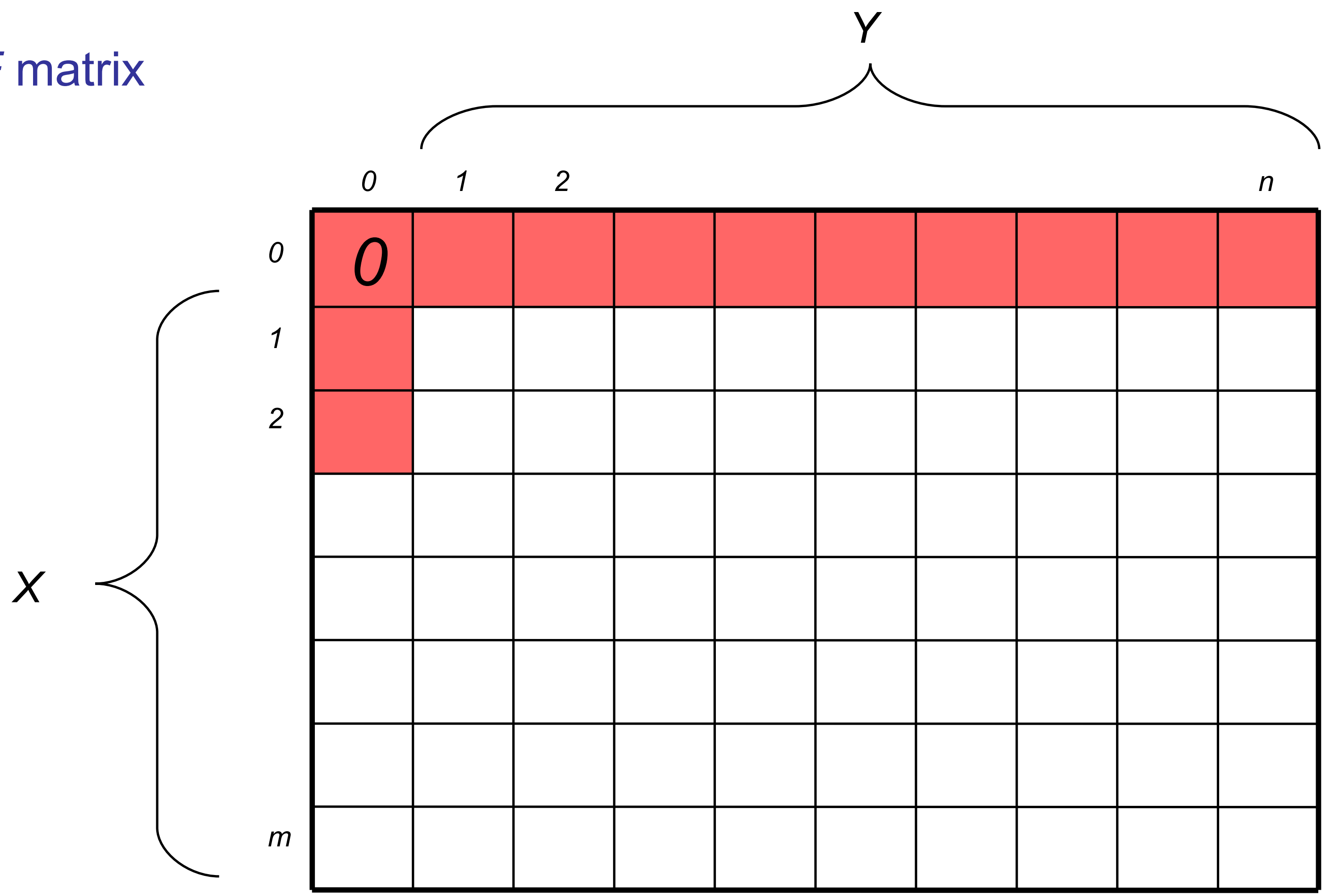
Filling up table

F matrix



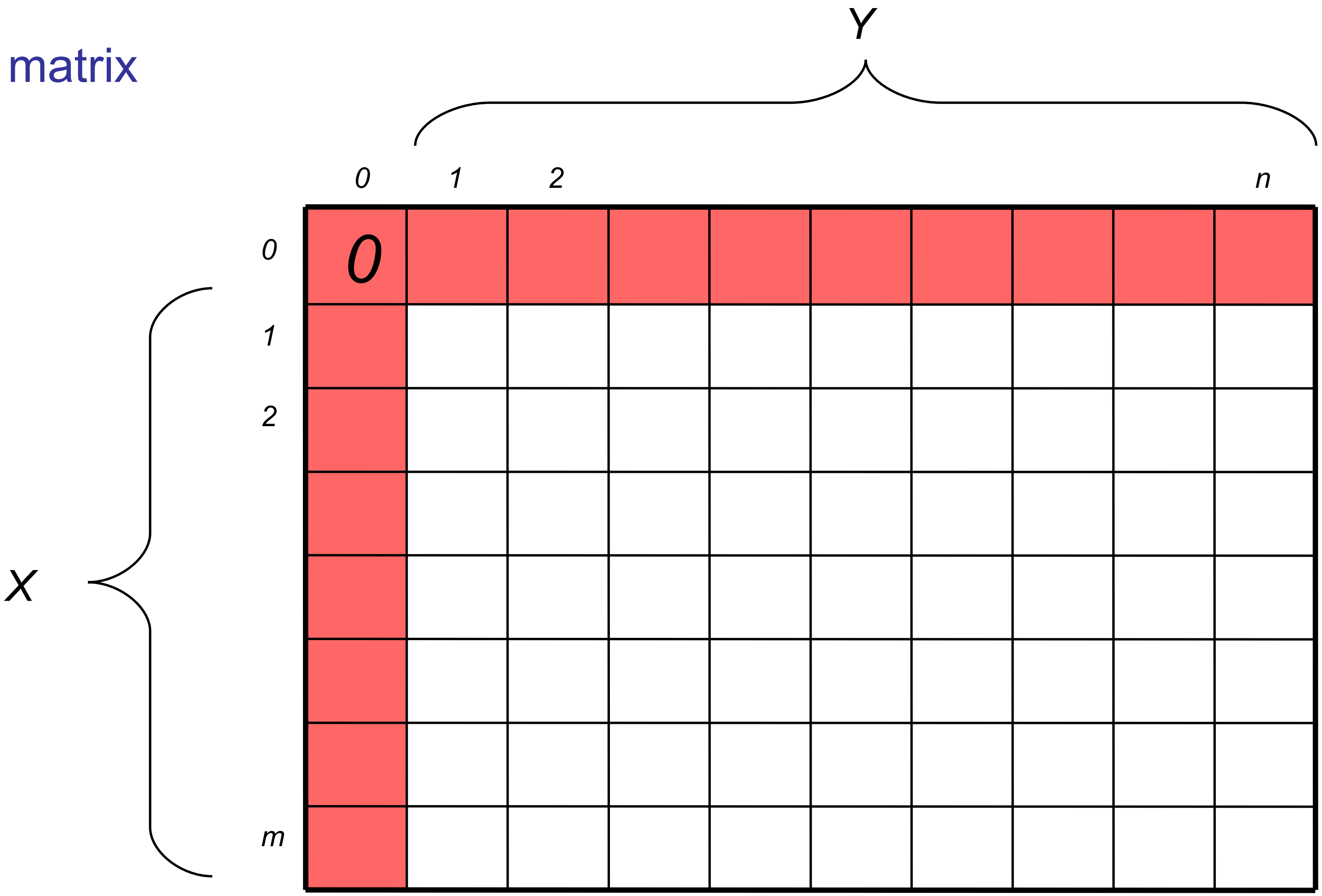
Filling up table

F matrix



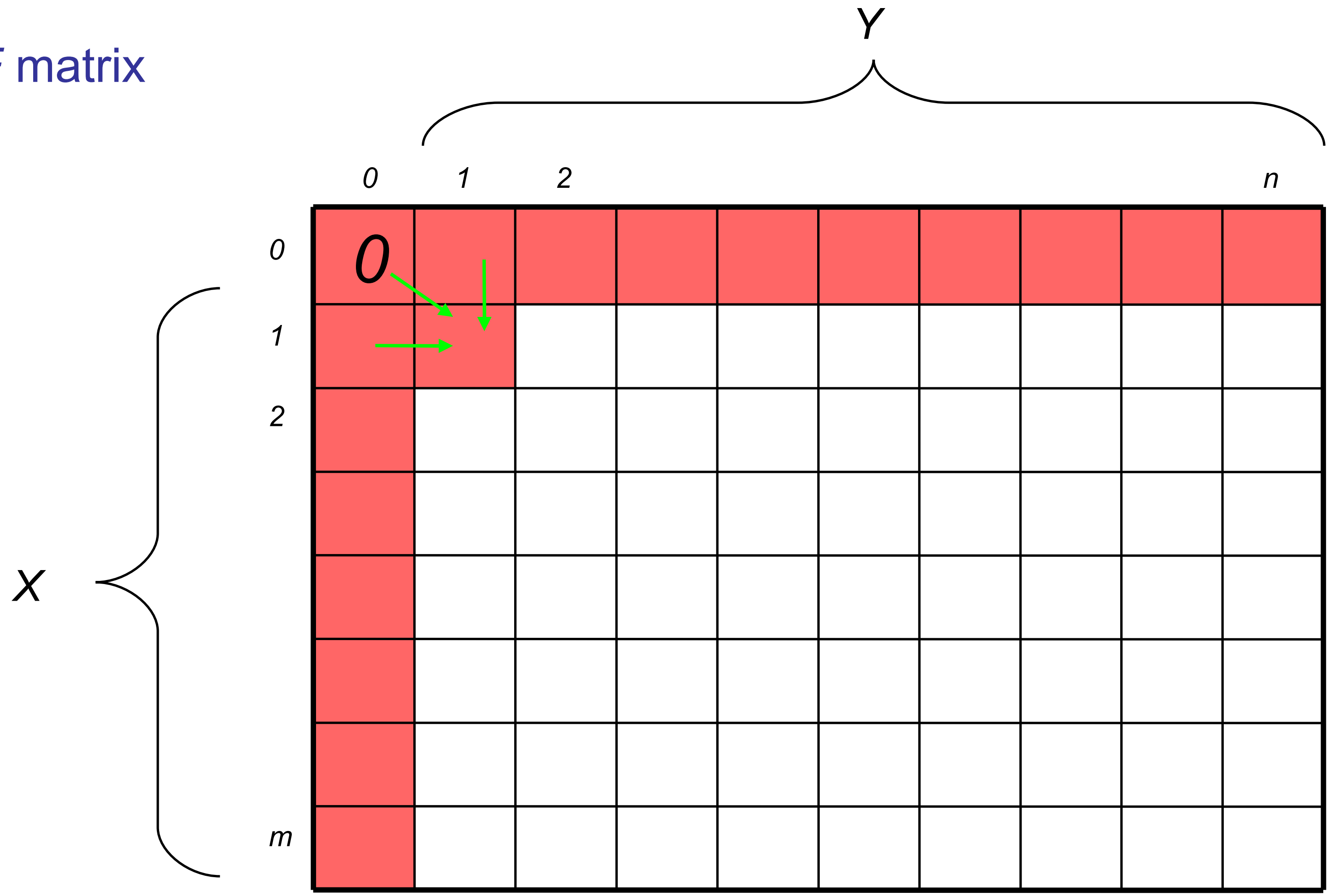
Filling up table

F matrix



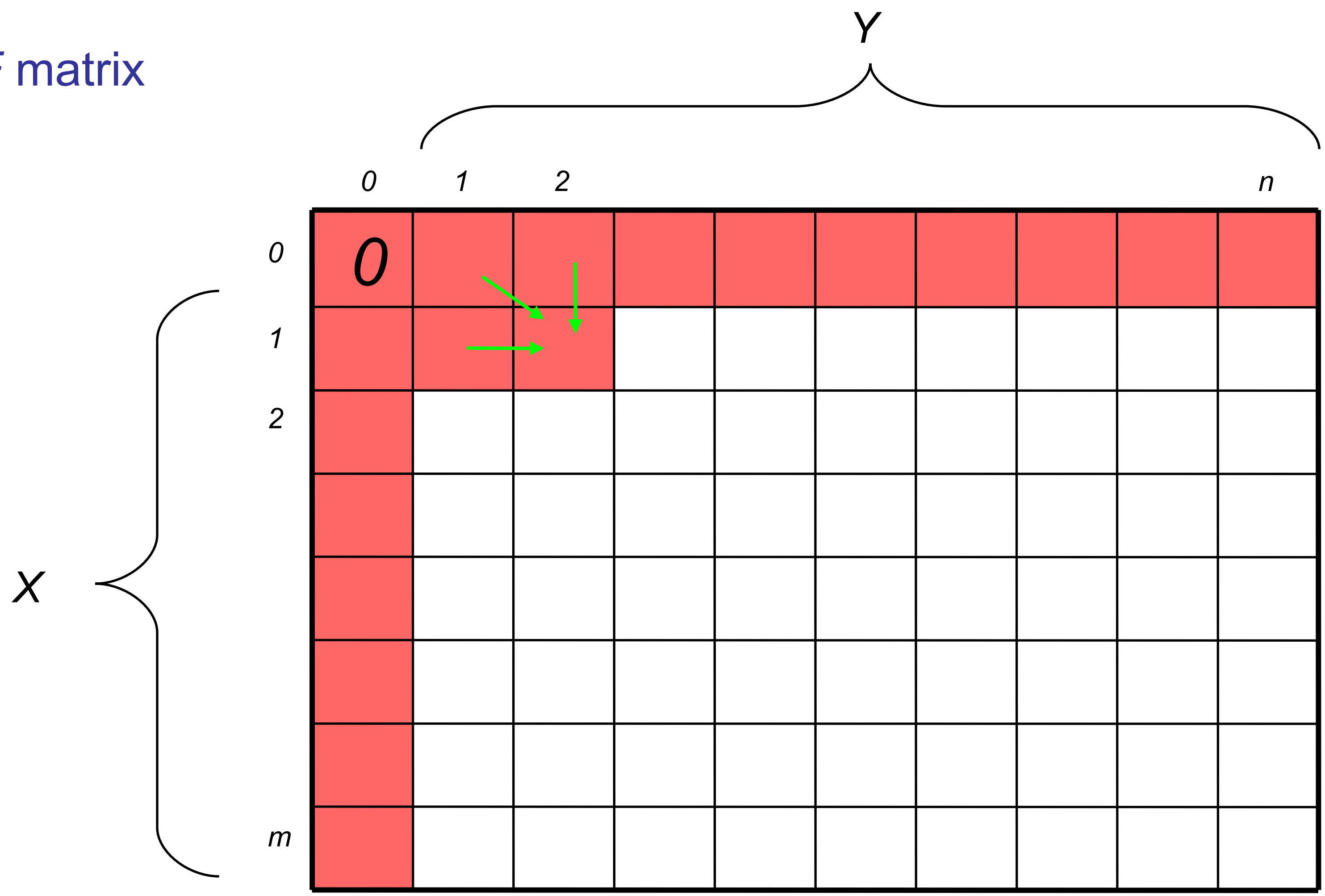
Filling up table

F matrix

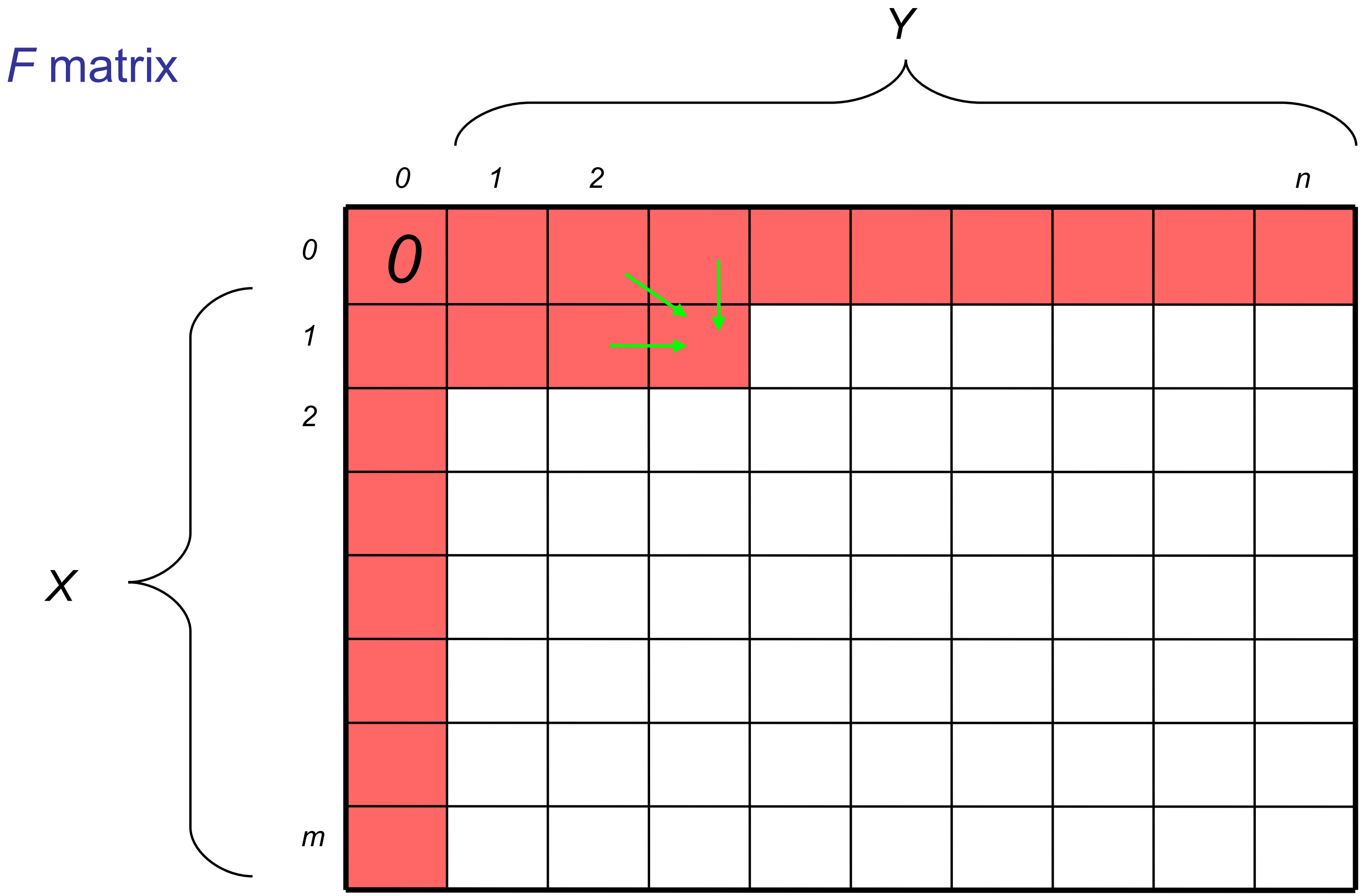


Filling up table

F matrix

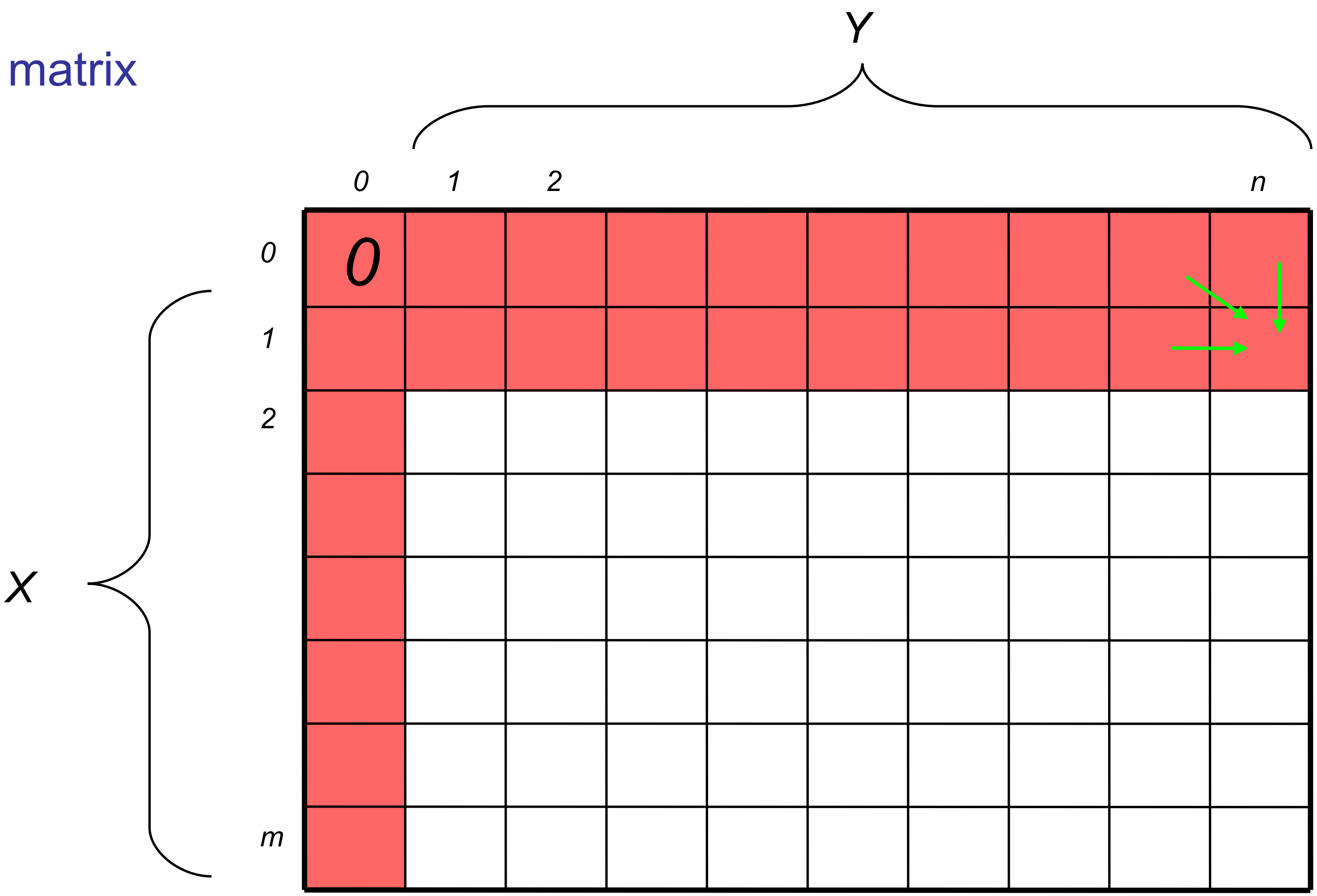


Filling up table

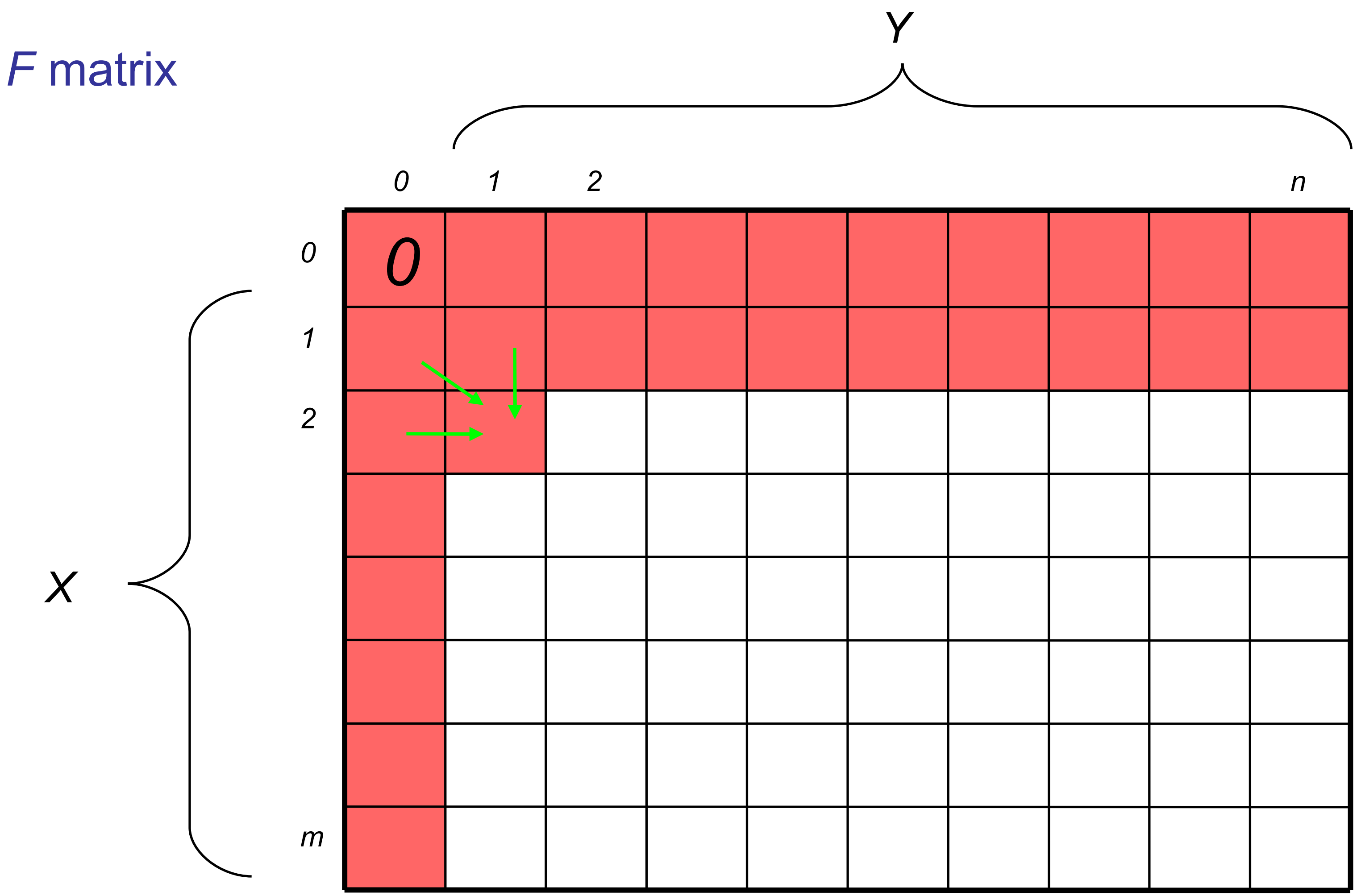


Filling up table

F matrix

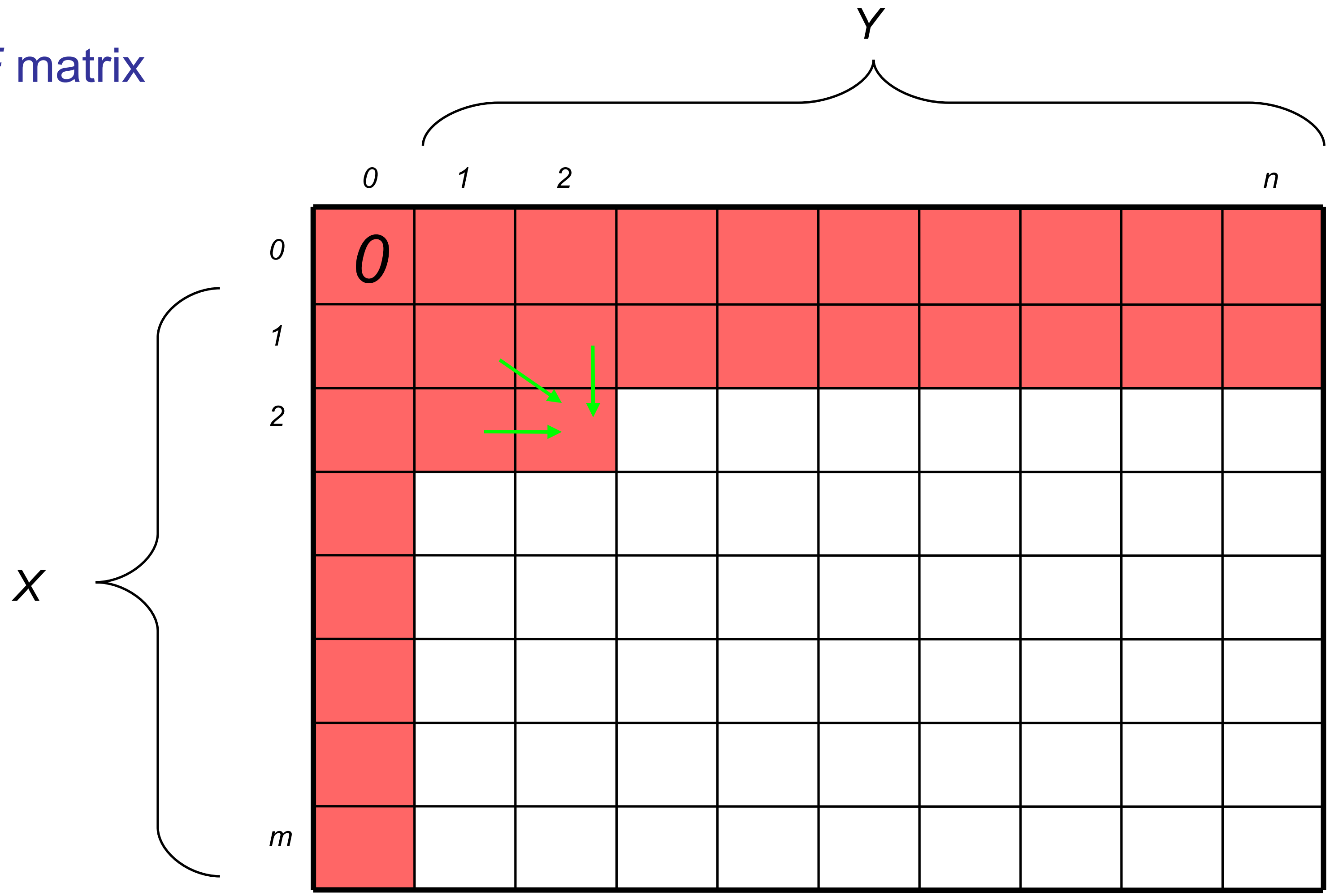


Filling up table

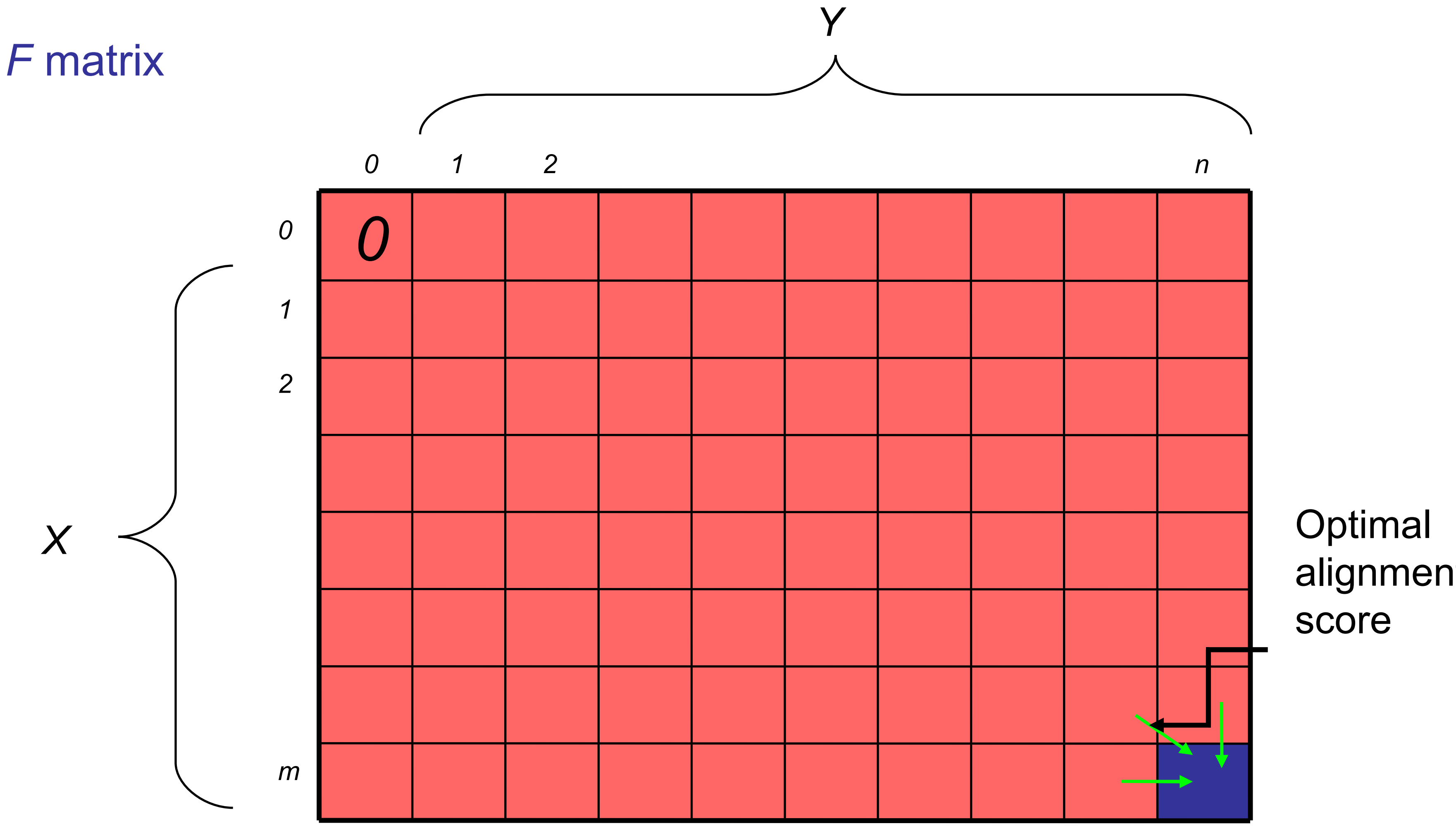


Filling up table

F matrix



Filling up table



Constructing alignment