- **Interleaving** → quando lo abbiamo fatto?

(i, j) $i > 0$ $j > 0$ $S_{i,j}$ = "W_{i+j} è interleaving di X_i e Y_j?"

se $W_{i+j} \neq X_i \wedge W_{i+j} \neq Y_j \Rightarrow S_{i,j} =$ **FALSE**. } CASO BASE

se $W_{i+j} = X_i \wedge W_{i+j} \neq Y_j \Rightarrow S_{i,j} = X_{i-1}, Y_j, W_{i+j-1}$

se $W_{i+j} \neq X_i \wedge W_{i+j} = Y_j \Rightarrow S_{i,j} = X_i, Y_{j-1}, W_{i+j-1}$

se $W_{i+j} = X_i \wedge W_{i+j} = Y_j \Rightarrow S_{ij} = S_{i-j} \vee S_{ij-1}$

ALGORITMO BOTTOM-UP:

$M_{m+1, n+1}$ → matrice per ogni coppia i, j.

```
INTERLEAVING (X, Y, W):
  m = length(X);
  n = length(Y);
  for (i = 0 to m):
      for (j = 0 to n):
          Caso 1 → if (i = j = 0):
                      Sij = true;
          caso 2 → if (i > 0 ∧ j = 0):
                      if (wi ≠ xi)
                          Sij = false;
                      else
                          Sij = Si-1,j;
          Caso 3 → if (i = 0 ∧ j > 0):
                      if (wi ≠ yj)
                          Sij = false;
                      else
                          Sij = Si,j-1;
```

```
caso 4 —▷ if (i>0 ∧ j>0):
              if (w_{i+j} ≠ x_i ∧ w_{i+j} ≠ y_j)
                  S_{ij} = false;
              if (w_{i+j} = x_i ∧ w_{i+j} ≠ y_j)
                  S_{ij} = S_{i-1,j}
              if (w_{i+j} ≠ x_i ∧ w_{i+j} = y_j)
                  S_{ij} = S_{i,j-1}
              if (w_{i+j} = x_i ∧ w_{i+j} = y_j)
                  S_{ij} = S_{i,j-1} ∨ S_{i-1,j}
return S.
```

Stringhe palindrome:

Σ alfabeto.

Sia $S = a_1 \dots a_n$ una stringa su Σ di lunghezza n.
$S = \epsilon$ stringa vuota ($n = 0$).

Si vuole determinare il numero minimo di caratteri da aggiungere ad S per renderla palindroma.

Stringa palindroma che contiene S.

Esempi:

$S = \epsilon$ è già palindroma $\longrightarrow \emptyset$
$S =$ "ADDA" è già palindroma $\longrightarrow \emptyset$
$S =$ "CASA" non è palindroma $\longrightarrow$ aggiungo ?

"CASAASAC" $\rightarrow$ 4
"CASAC" $\rightarrow$ 1
"ACASACA" $\rightarrow$ 3

$\rightarrow$ posso aggiungere in vari modi, ma il migliore è il 2°.

Formulazione del problema:

Sia $f : \Sigma^* \longrightarrow \mathbb{N}$ definito come segue:

$\forall s \in \Sigma^*$ $f(s) =$ numero minimo di caratteri da aggiungere a S per ottenere una stringa palindroma.

PROBLEMA: (usare questo formato x def. il probl.)

Istanza $s \in \Sigma$

Soluzione $f(s)$

SOTTOPROBLEMI:

$S = \epsilon \Rightarrow f(s) = 0$

$S = a \Rightarrow f(s) = 0$

↓

qualsiasi $a \in \Sigma$

Caso base.

Se S é composta da almeno 2 caratteri:

$S = a S' b \qquad |S'| = |S| - 2$

Dove a é il carattere iniziale di S b è il carattere finale di S. S' è una stringa di lunghezza < S.

Ora: Casi passo

- Se $a = b$ S è palindroma, se S' è palindroma.

$$f(s) = f(s')$$

- Se $a \neq b$

$$\left.\begin{array}{l} Sa = aS'ba \\ bS = baS'b \end{array}\right\} \quad 1 + \min\{f(aS'), f(S'b)\}$$

con variabili:

$i =$ primo carattere stringa;

$j =$ ultimo carattere stringa;

$S_{i,j}$:

Lunghezza sottostringa partendo dall' i-esimo ed arrivando al j-esimo carattere della stringa originale.

$\downarrow$

$S_{1,n} = a_1 \dots a_n$

(i,j) con $i \in \{1 \dots n\}$ $j \in \{1 \dots n\}$

- quando $i > j$ $S_{i,j} = \epsilon$;
- quando $i = j$ $S_{i,j} = a_i$;

$\rightarrow$ **casi base.** $m_{ij} = 0.$

Chiamo $m_{ij} = f(S_{ij})$ (minimo x pal).

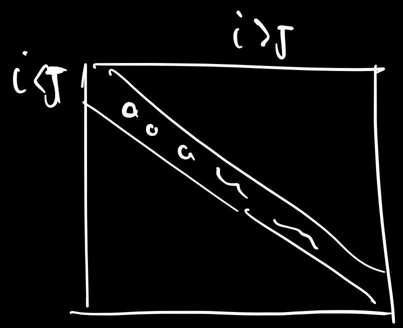
Casi passo:

- quando $i < j \wedge i = j$ $S_{i,j} = S_{i+1,j-1}$ $m_{i+1,j-1}$
- quando $i < j \wedge i \neq j$

$$S_{i,j} = 1 + \min\{ a_i \mid S_{i,j} ; S_{i,j} \mid b_j \}$$

Soluzione: m_{1n}

* Provare a riempire la $M_{n \times n}$.



etc...

(guarda da solo)