Esercizio: Shortest Palindrome Completion

Def.: una stringa $Z = \langle z_1, z_2, \dots, z_m \rangle$ è <u>palindroma</u> se
$$z_{1+h} = z_{m-h} \quad \forall\, 0 \le h \le m-1$$

esempi: abba, aba

Problema: data $X = \langle x_1, x_2, \dots, x_n \rangle$, un <u>completamento a palindromo (CP)</u> di X è una stringa $Z = \langle z_1, z_2, \dots, z_m \rangle$, $m \geq n$, con
- Z è palindroma
- X è sottosequenza di Z

esempio: $Z = abccba$ è CP di $X = acb$

Determina il CP di X di lunghezza minima

Suggerimento: spazio dei sottoproblemi: $X_{i \dots j}$
algoritmo quadratico in n

$\rightarrow$ bisogna, $\forall\, i,j \quad 1 \le i \le j \le n$ determinare il minimo CP di $X_{i \dots j}$

Casi base: $i = j \qquad \langle x_i \rangle$
$$CP(X_{i \dots i}) = X_{i \dots i} \qquad \left| CP(X_{i \dots i}) \right| = 1$$

$j = i+1$
$$X_{i \dots i+1} = \langle x_i, x_{i+1} \rangle$$
se $x_i = x_{i+1} \rightarrow CP(X_{i \dots i+1}) = X_{i \dots i+1}$
se $x_i \neq x_{i+1} \rightarrow$ un possibile $CP(X_{i \dots i+1})$ è $\langle x_i, x_{i+1}, x_i \rangle$

Caso generale: $l = j - i + 1 > 2$
$$X_{i \dots j} = \langle x_i, x_{i+1}, \dots, x_j \rangle$$

caso $x_i = x_j \rightarrow Z = CP(X_{i \dots j})$ ottimo

$z_1 = z_k = x_i$ e

$Z_{2 \dots k-1} = CP(X_{i+1 \dots j-1})$ ottimo

caso $x_i \neq x_j \rightarrow z_1 = z_k = x_i$ oppure $z_1 = z_k = x_j$

$Z_{2 \dots k-1} = CP(X_{i+1 \dots j})$ ottimo

$Z_{2 \dots k-1} = CP(X_{i \dots j-1})$ ottimo

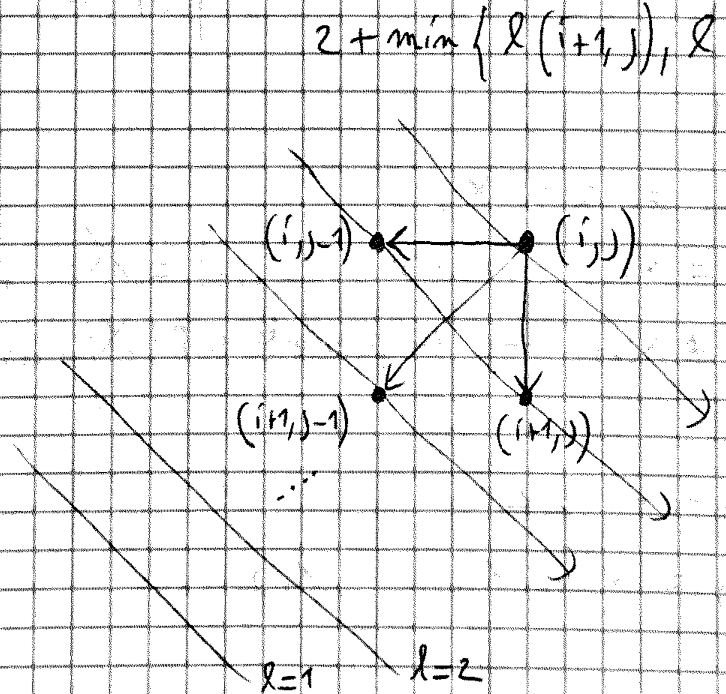
Esercizio: dimostra questa proprietà di sottostruttura ottima

Riccorrenza sulle lunghezze:

chiamo $\ell(i,j) = |CP(X_{i \dots j})|$

$$\ell(i,j) = \begin{cases} 1 & i=j \\ 2 & j=i+1 \text{ e } x_i = x_j \\ 3 & j=i+1 \text{ e } x_i \neq x_j \\ 2+\ell(i+1,j-1) & j>i+1 \text{ e } x_i = x_j \\ \min\{2+\ell(i+1,j),\ 2+\ell(i,j-1)\} & j>i+1 \text{ e } x_i \neq x_j \end{cases}$$

$$\min\{2+\ell(i+1,j),\ 2+\ell(i,j-1)\} = 2+\min\{\ell(i+1,j),\ \ell(i,j-1)\}$$



```
SHORTEST_PALINDROME_COMPLETION (X)
    n ← length (X)
    for i ← 1 to n-1 do                    \\inizializza le 2 diagonali
        L[i, i] ← 1
        if x_i = x_{i+1} then
            L[i, i+1] ← 2
        else L[i, i+1] ← 3
    L[n, n] ← 1
    for l ← 3 to n do
        for i ← 1 to n-l+1 do
            j ← i+l-1
            if x_i = x_j then
                L[i, j] ← 2 + L[i+1, j-1]
            else L[i, j] ← 2 + MIN(L[i+1, j], L[i, j-1])
    return L[1, n]
```

Complessità: $\sum_{i=1}^{n-1} 1 + \sum_{l=3}^{n} \sum_{i=1}^{n-l+1} 1 = n-1 + \sum_{l=3}^{n} (n-l+1) =$

$$= n-1 + n-2 + \sum_{j=1}^{n-3} j =$$

$$= \sum_{j=1}^{n-1} j = \frac{n(n-1)}{2}$$

Esercizio: introduci nel codice il calcolo dell'informazione addizionale, e scrivi un algoritmo ricorsivo PRINT_PC (i, j, B, X) che stampi PC ($X_{i \ldots j}$).