Titolo unità didattica: Stringhe ed elaborazione di testi [09]

Titolo modulo : Operazioni elementari su stringhe [01-T]

Le stringhe di caratteri: operazioni di concatenazione, confronto,..

Argomenti trattati:

- ✓ stringhe di caratteri su un alfabeto
- ✓ stringhe e sottostringhe
- ✓ operazioni di base su stringhe

Prerequisiti richiesti: P1-07-11-T

stringhe di caratteri

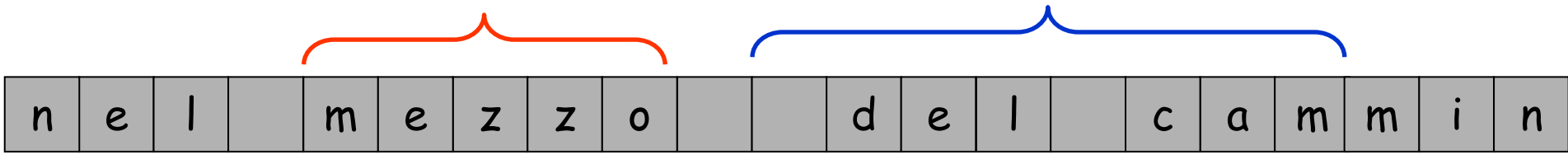
- ✓una **stringa** su un alfabeto $\{c_j\}_{j=1,n}$ è una sequenza di caratteri dell'alfabeto:

  $$c_1c_2c_3c_4c_5\ldots c_k$$

- ✓ una stringa è costituita da un **numero finito** di caratteri, che è la **lunghezza** della stringa
  ($c_1c_2c_3c_4c_5\ldots c_k$ ha lunghezza k)
- ✓ la **stringa vuota** è la stringa di lunghezza zero (0-stringa)
- ✓una stringa di lunghezza k è detta **k-stringa**
- ✓ in una stringa, ogni carattere ha sua una posizione d'ordine che si chiama **indice** di stringa
- ✓ una **sottostringa** è una *porzione* di una stringa, cioè una sequenza di caratteri **consecutivi** di una stringa

stringhe di caratteri

- ✓una **sottostringa** è una *porzione* di una stringa, cioè una sequenza di caratteri **consecutivi** di una stringa
- ✓ una sottostringa di lunghezza `p`, il cui **primo** carattere si trova nella posizione di indice `i` della stringa è detta `p`-sottostringa di inizio `i`



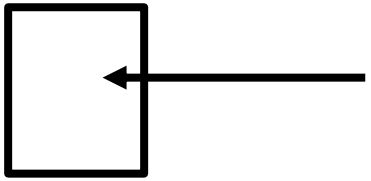
21-stringa, stringa di lunghezza 21

5-sottostringa di inizio 5

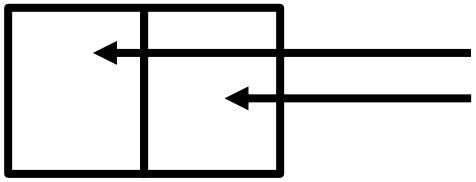
8-sottostringa di inizio 11

stringhe di caratteri

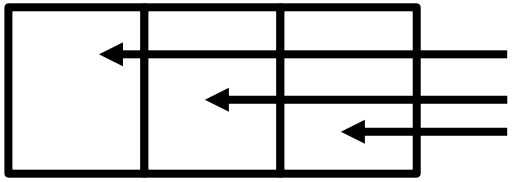
- ✓l'insieme di tutte le **1**-stringhe su un alfabeto di **n** caratteri ha cardinalità **n**
- ✓ l'insieme di tutte le **2**-stringhe su un alfabeto di **n** caratteri ha cardinalità $\mathbf{n^2}$
- ✓ l'insieme di tutte le **k**-stringhe su un alfabeto di **n** caratteri ha cardinalità $\mathbf{n^k}$



n caratteri a disposizione => **n** **1**-stringhe



n caratteri a disposizione => $\mathbf{n^2}$ **2**-stringhe



n caratteri a disposizione => $\mathbf{n^3}$ **3**-stringhe

stringhe di caratteri

- ✓l'insieme di tutte le **1**-stringhe su un alfabeto di **n** caratteri ha cardinalità **n**
- ✓ l'insieme di tutte le **2**-stringhe su un alfabeto di **n** caratteri ha cardinalità **n^2**
- ✓ l'insieme di tutte le **k**-stringhe su un alfabeto di **n** caratteri ha cardinalità **n^k**

Esempio: alfabeto $\{a,b\}$ n = 2

k = 1 a ; b

k = 2 aa ; ab ; ba; bb

k = 3 aaa ; aab ; aba; abb; baa; bab; bba; bbb

stringhe di caratteri

- ✓una **permutazione** di una **k**-stringa è una **k**-stringa composta dagli stessi caratteri

Esempio permutazioni:

3-stringa abc

abc, acb, bac, bca, cab, cba

stringhe di caratteri

- ✓una **permutazione** di una **k**-stringa è una **k**-stringa composta dagli stessi caratteri
- ✓ l'insieme di tutte le permutazioni di una **k**-stringa ha cardinalità **k!**

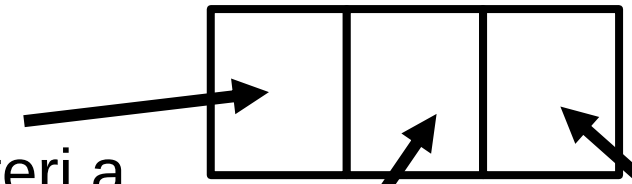
Esempio permutazioni:

stringa abc

k = 3

$$3 \cdot 2 \cdot 1 = 3! = 6$$

permutazioni



quanti caratteri a disposizione ?

3

quanti caratteri a disposizione ?

2

quanti caratteri a disposizione ?

1

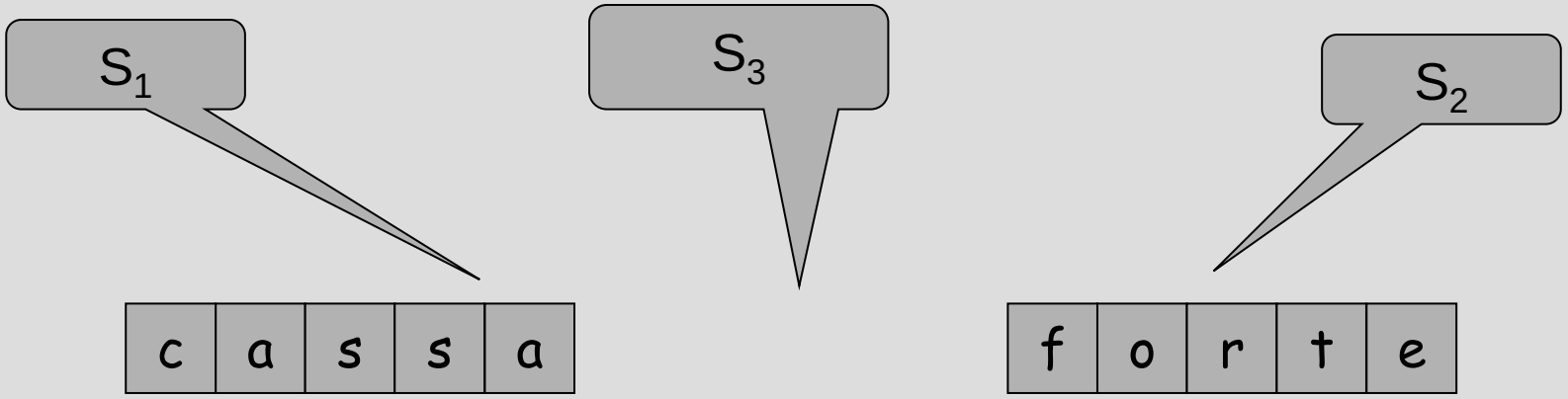
operazioni di base su **stringhe** di caratteri

- **concatenazione**
- **ricerca di** un carattere in una stringa
- **confronto** tra due stringhe
- **matching** di due stringhe
- creazione di una **copia** di una stringa
- calcolo della **lunghezza** di una stringa
- **confronto di sottostringhe** di due stringhe
- **copia di una sottostringa** di una stringa
- **ricerca di una sottostringa** in una stringa

concatenazione (**cat**) di stringhe

- stringhe:

S_1 = cassa S_2 = forte



- si ottiene una nuova stringa S_3 che è la concatenazione delle due stringhe S_1 ed S_2

$S_3 = cat(S_1, S_2)$ cassaforte

ricerca di un carattere in una stringa

- ricercare (ricerca sequenziale) un dato carattere (chiave) all'interno di una stringa data
- ricercare (ricerca sequenziale) un dato carattere (chiave) all'interno di una certa sottostringa di una stringa data

confronto fra due stringhe

- determinare l'uguaglianza o la diseguaglianza tra due stringhe date. Confronto tra caratteri di uguale posizione:

S_1 = cassaforte S_2 = pianoforte

cmp(S_1,S_2) false

confronto fra due stringhe fra due stringhe per determinare quale delle due stringhe precede l'altra nell'ordine alfabetico

- confronto tra caratteri di uguale posizione:

S_1 = cassaforte S_2 = casseforti

cmpalf(S_1,S_2) S_1

matching fra due stringhe

- determinare il numero di caratteri (di ugual posto) delle due stringhe che risultano uguali

S_1 = cassaforte S_2 = pianoforte

match(S_1,S_2) 5

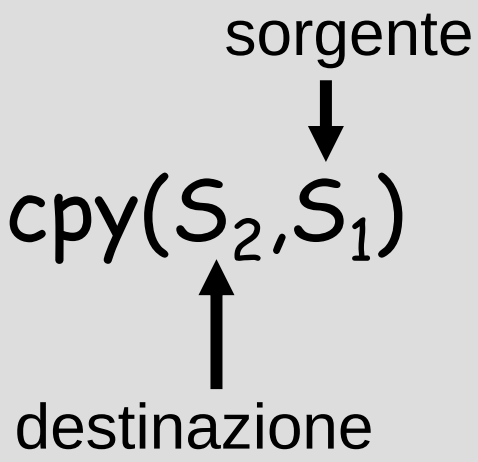
punteggio del matching 5

creazione di una **copia** di una stringa

- assegnare la stringa sorgente S_1 alla striga destinazione S_2

stringhe:

S_1 = cassaforte S_2 = NULL (stringa vuota)



calcolo della **lunghezza** di una stringa

- lunghezza di una stringa, ovvero numero dei caratteri che costituiscono la stringa

S = cassaforte

len(S) 10

confronto di sottostringhe

- stringhe:

S_1 = cassaforte S_2 = pianoforte

Esempio:

verificare l'uguaglianza tra le sottostringhe di lunghezza `m` e inizio 1, `m`=5

ncmp(S_1,1,S_2,1,m) false

inizio 6 lunghezza 5

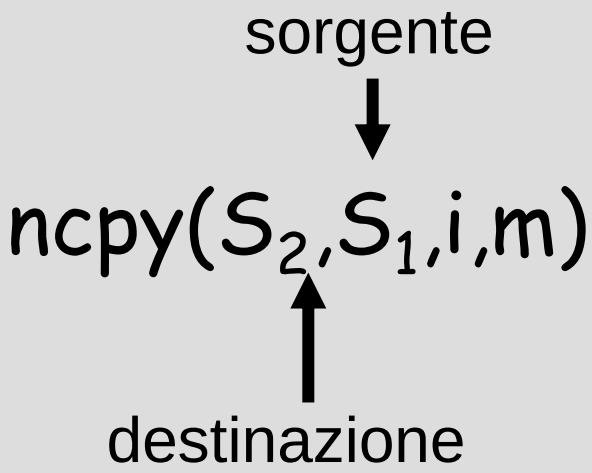
ncmp(S_1,6,S_2,6,m) true

inizio 1 di S_1, inizio 4 di S_2, lunghezza 5

ncmp(S_1,1,S_2,4,m) false

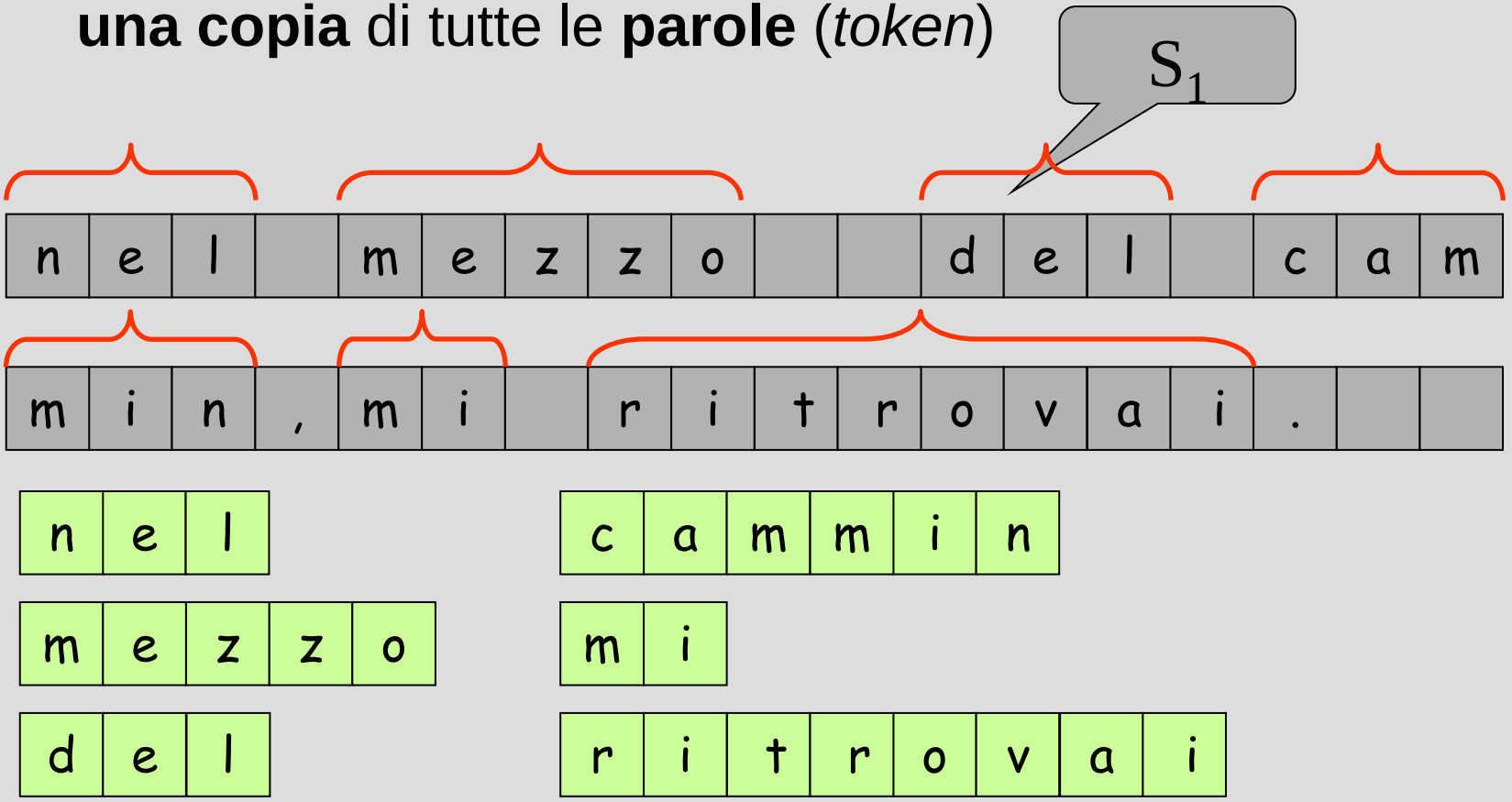
copia di una sottostringa

- data una stringa S_1, creare una copia S_2 della sottostringa di lunghezza `m` e inizio i



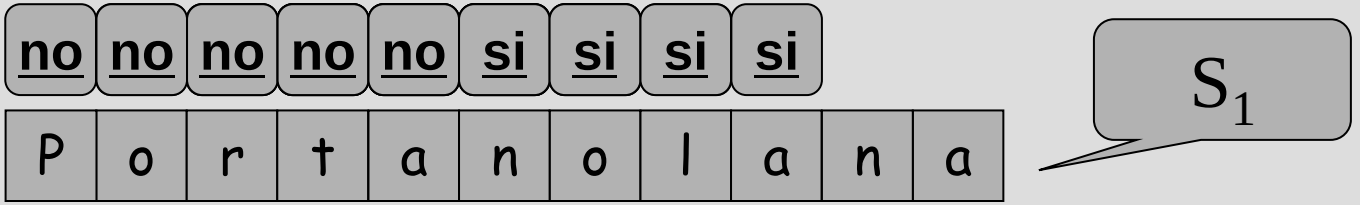
estrazione di **token** da una stringa (**tok**)

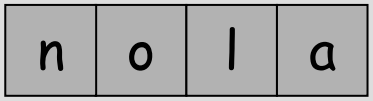
- data una stringa S_1, che contiene un testo, cioè un insieme di parole separate da **separatori** (spazio, segni di punteggiatura, andata a capo, etc..), **estrarre una copia** di tutte le **parole** (*token*)



ricerca di una sottostringa in una stringa (*string matching, pattern matching*)

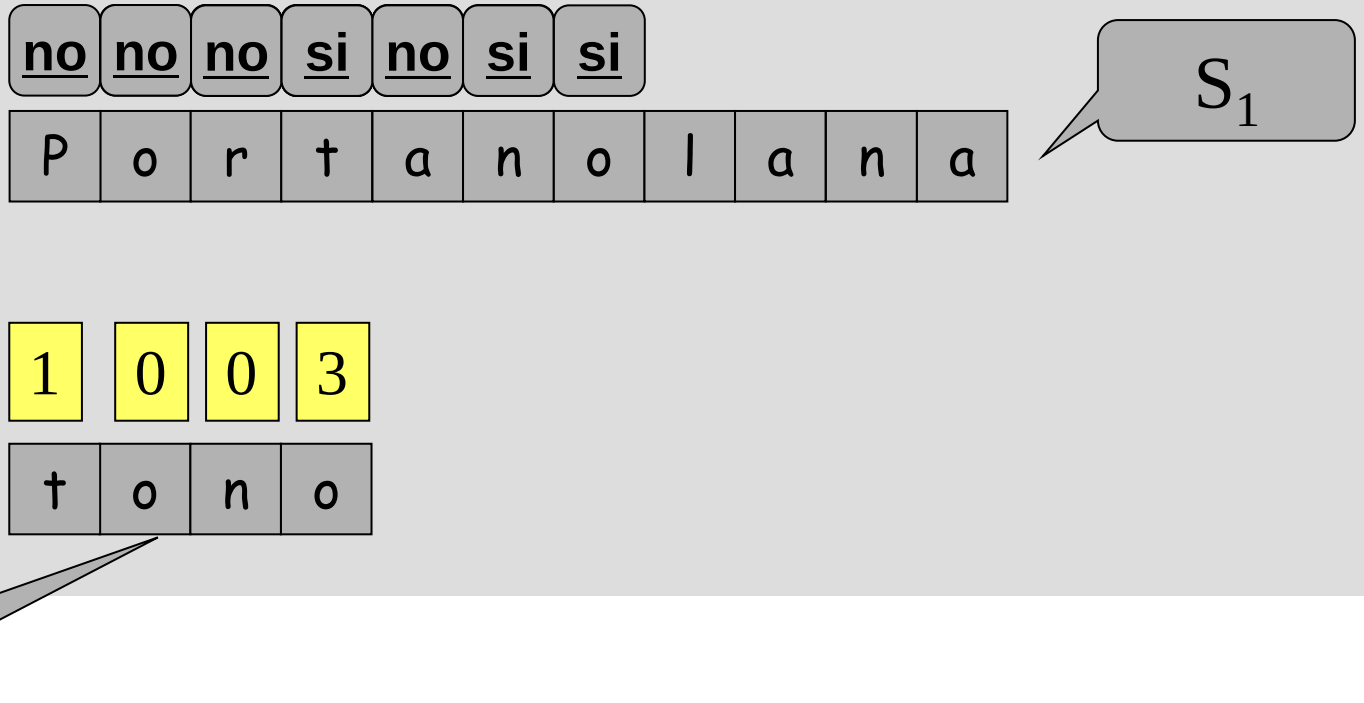
- date due stringhe S_1 e S_2, ricercare le occorrenze della stringa S_2 come sottostringa della stringa S_1





ricerca di una sottostringa in una stringa che più si *avvicina* una stringa data
(*best matching, matching migliore*)

- date due stringhe S_1 e S_2, trovare la sottostringa della stringa S_1 che ha il maggior numero di caratteri (di ugual posto) in comune con la stringa S_2



ricerca di una sottostringa in una stringa (*string matching, pattern matching*)

- ✓ sapere se una stringa (chiave) appare come sottostringa di un'altra stringa
- ✓ contare il **numero delle occorrenze** di una sottostringa in una stringa
- ✓ sapere l'**indice** di una sottostringa in una stringa
- ✓ **ricercare** una sottostringa in una stringa e **sostituirla** con un'altra sottostringa (*find/replace*)

pattern matching in Bioinformatica

- ✓ il DNA si trova nel nucleo delle cellule
- ✓ è il *programma* che ogni cellula del nostro corpo (ce ne sono circa 10^{13}) deve continuamente eseguire
- ✓ l'elemento base del DNA è una molecola chiamata **nucleotide**
- ✓ per ogni specie o essere vivente esistono **solo 4 nucleotidi**, identificati dai nomi

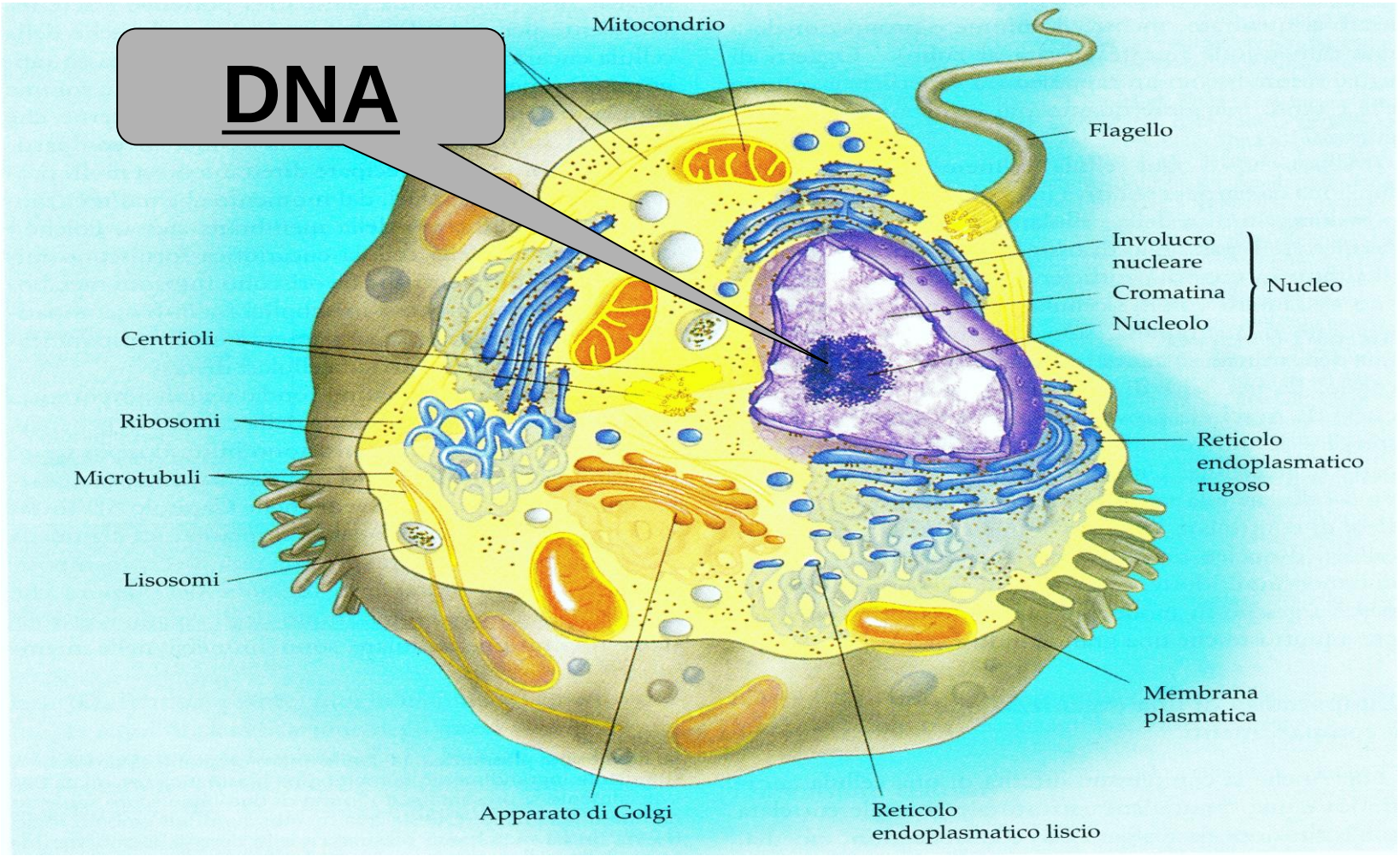
Adenina, **T**imina, **C**itosina, **G**uanina

curiosità

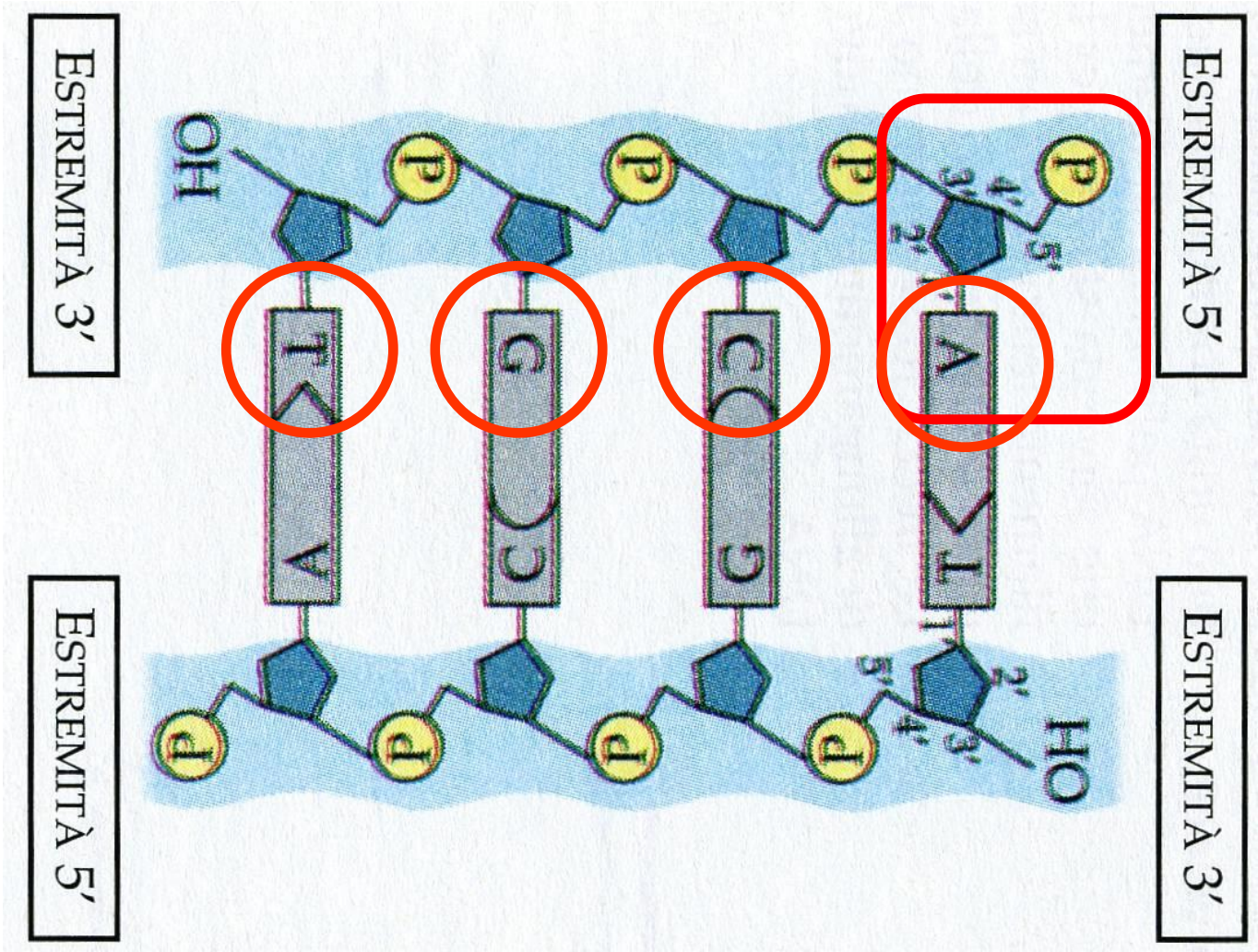
- ✓ nell'uomo il DNA è organizzato (suddiviso) in 23 coppie di **cromosomi**
- ✓ solo il cromosoma 23 è diverso tra maschio e femmina
- ✓ in ogni coppia di cromosoma un elemento proviene dal padre e uno dalla madre
- ✓ srotolando il DNA di una singola cellula si avrebbe un nastro di circa 2 metri di lunghezza
- ✓ mettendo in fila i nastri di tutte le cellule di un uomo si avrebbe un nastro lungo 200 miliardi di Km (più di 1000 volte la distanza Terra-Sole)
- ✓ considerando il DNA di una cellula come una stringa sull'alfabeto {A,C,G,T}, la sua lunghezza è di circa 3 miliardi di caratteri

pattern matching in Bioinformatica

cellula

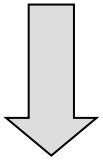


pattern matching in Bioinformatica



pattern matching in Bioinformatica

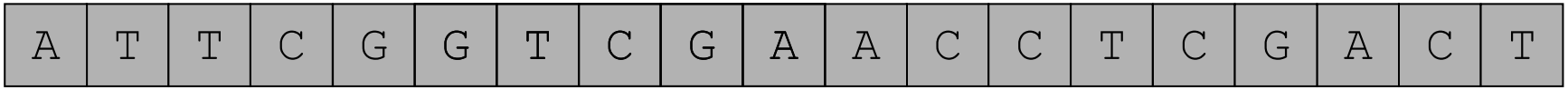
- ✓ il DNA è una lunga catena (doppia) di nucleotidi



il DNA è un **testo** (**stringa**) sull'alfabeto {A,T,C,G}

pattern matching in Bioinformatica

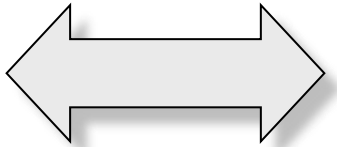
- un **gene** è un segmento di DNA, che contiene l'informazione relativa a una specifica funzionalità
- il DNA è un testo (**stringa**) contenente tutte le informazioni necessarie per la vita della cellula e dell'individuo
- l'insieme di tutte le informazioni genetiche, e quindi dell'intero DNA di un individuo o di una specie, è detto **genoma**



gene (sottostringa)

pattern matching in Bioinformatica

ricercare un **particolare gene** in una **sequenza di DNA**



ricercare un **pattern** in una **stringa**