

Tutoraggio Ricerca Operativa 2021/2022

Esercizi vari

Aurora Rossi, Alice Raffaele, Romeo Rizzi

Università degli Studi di Verona

21 aprile 2022

Rank ed unrank di stringhe su un alfabeto

- ① **Stringa**: sequenza finita (di lunghezza s) di caratteri presi da un qualche alfabeto di riferimento.
- ② **Rank**: classificare nel nostro problema è disporre le stringhe in ordine lessicografico.
- ③ **Unrank**: declassificare, partendo da una stringa trovare il suo posto all'interno di tutte le liste.

Rank ed unrank di stringhe su un alfabeto

Vediamo un esempio (alfabeto: $\{A, B\}$, lunghezza stringa=3) per chiarire :

stringa 0: AAA

stringa 1: AAB

stringa 2: ABA

stringa 3: ABB

stringa 4: BAA

stringa 5: BAB

stringa 6: BBA

stringa 7: BBB

Richieste:

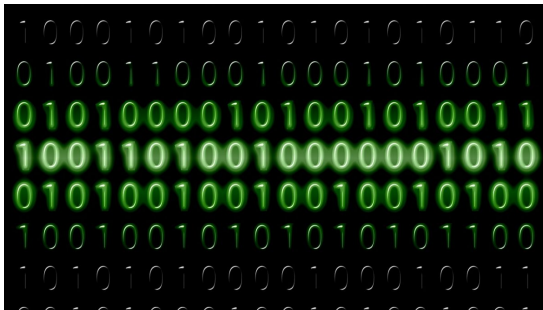
- ➊ **Unrank**: data una posizione, stampare la stringa corrispondente (posizione 0 stringa AAA).
- ➋ **Rank**: data una stringa, stampare la posizione corrispondete .

Rank ed unrank di stringhe su un alfabeto

Domande:

- 1 Dato un alfabeto di lunghezza s quante stringhe ci aspettiamo?
- 2 Prova a implementare i codici di rank e unrank con python.
- 3 Cosa cambia se l'alfabeto avesse lunghezza maggiore di 2?

Problema: ricevi in input un numero n (da convertire in numero binario) e devi trasformarlo nel numero 0 impiegando il minor numero possibile di mosse .



Sistema binario

Un **numero binario** è una sequenza di cifre binarie (dette bit). Ogni cifra in posizione n (contate da destra verso sinistra iniziando da 0) si considera moltiplicata per 2^n , anziché per 10^n , come avviene nella numerazione decimale.

La formula per convertire un numero da binario a decimale (dove con d_n si indica la cifra di posizione n all'interno del numero, contate da destra verso sinistra iniziando da 0) è

$$d_n 2^n + d_{n-1} 2^{n-1} + \dots + d_1 2^1 + d_0 2^0 = N_{10}$$

Ad esempio

$$1001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9_{10}.$$

Task 1: implementare un codice Python che trasformi un numero da decimale a binario.

Mosse a disposizione:

- 1 inverte il valore del bit più a destra, ossia il bit di parità (quello meno significativo)
- 2 inverte il valore del bit alla immediata sinistra del bit posto più a destra tra quelli settati ad uno.

Task 2: implementare una funzione che ritorna il minimo numero di mosse per passare da n a zero.

Task 3: implementare una funzione che ritorna 1 oppure 2: la risposta $i (= 1, 2)$ è corretta se la mossa i , applicata ad n , ci porta un passo più vicini allo zero.