

## Esercitazione 2

- Sequenze numeriche in vettore (Lab 0, es. 1)
- Rotazione di vettori (Lab 0, es. 3)
- Iterazione su matrici (Lab 0, es. 4)
- Operazioni su insiemi (“Algoritmi e programmazione in pratica: da specifiche a codice C”, sez. 1.1.4)

# Sequenze numeriche in vettore

Sia  $V$  un vettore di  $N$  interi ( $N \leq 30$ ). Si scriva un programma in C che, una volta acquisito da tastiera tale vettore, visualizzi tutti i sottovettori di dimensione massima formati da celle contigue contenenti dati non nulli.

- Problema di *ottimizzazione*
- Enumerazione delle soluzioni ammissibili
- Valutazione della funzione obiettivo
- In questo caso, sono richieste tutte le soluzioni ottime

# Sequenze numeriche in vettore

Soluzione 1: enumerazione delle sequenze di lunghezza  $l = n, n - 1, \dots, 1$ , con verifica di ammissibilità. Le soluzioni ottime sono date dalle soluzioni ammissibili con  $l$  massimo

# Sequenze numeriche in vettore

Soluzione 1: enumerazione delle sequenze di lunghezza  $l = n, n - 1, \dots, 1$ , con verifica di ammissibilità. Le soluzioni ottime sono date dalle soluzioni ammissibili con  $l$  massimo

Soluzione 2: identificazione della lunghezza massima di sequenze ammissibili tramite scansione lineare del vettore. Noto il massimo, una seconda scansione permette di stampare tutte le soluzioni ottime

# Sequenze numeriche in vettore

Soluzione 1: enumerazione delle sequenze di lunghezza  $l = n, n - 1, \dots, 1$ , con verifica di ammissibilità. Le soluzioni ottime sono date dalle soluzioni ammissibili con  $l$  massimo

Soluzione 2: identificazione della lunghezza massima di sequenze ammissibili tramite scansione lineare del vettore. Noto il massimo, una seconda scansione permette di stampare tutte le soluzioni ottime

Soluzione 3 (due varianti): utilizzo di un vettore ausiliario per memorizzare inizio sequenze ottime

Si scriva una funzione C in grado di permettere all'utente di far ruotare verso destra o verso sinistra i contenuti di un vettore di  $N$  interi, di un numero a scelta di posizioni  $P$ .

Soluzione 1:

- Rotazione a sinistra e a destra gestite esplicitamente
- Si effettuano  $P$  rotazioni di 1 posizione
- Algoritmo in loco, complessità  $O(NP)$

## Soluzione 2:

- Rotazione a sinistra e a (destra) gestite esplicitamente
- Si salvano i primi (ultimi)  $P$  elementi in un vettore temporaneo
- Si spostano i restanti  $N - P$  elementi a sinistra (a destra) di  $P$  posizioni
- Si ricopiano in coda (in testa) i  $P$  elementi salvati nel vettore temporaneo
- Richiede un vettore temporaneo di dimensione  $P$ , complessità  $O(N)$

## Soluzione 3:

- Come la soluzione 2, ma si unificano le operazioni di rotazione a destra e a sinistra, parametrizzando le posizioni di partenza e destinazione delle copie e degli spostamenti

## Soluzione 4:

- Si unificano spostamenti a destra e sinistra sfruttando l'osservazione che una rotazione di  $P$  posizioni verso destra corrisponde ad una rotazione di  $N - P$  posizioni verso sinistra

## Soluzione 5:

- Si sfruttano le proprietà dell'aritmetica modulare per eliminare la necessità di utilizzo di un vettore temporaneo

Un file di testo contiene una matrice di interi con il seguente formato:

- la prima riga del file specifica le dimensioni della matrice (numero di righe  $nr$  e numero di colonne  $nc$ ). Si assume che entrambi i valori siano comunque al più pari a 20
- ciascuna delle  $nr$  righe successive contiene gli  $nc$  valori corrispondenti a una riga della matrice, separati da uno o più spazi.

Si scriva un programma C che:

- legga tale matrice dal file di ingresso, il cui nome (massimo 20 caratteri) sia letto da tastiera
- chieda ripetutamente all'utente un valore dim compreso tra 1 e il minimo tra  $nr$  e  $nc$  e stampi tutte le sottomatrici quadrate di tale dimensione contenute nella matrice
- termini l'iterazione se l'utente inserisce un valore non coerente con le dimensioni della matrice
- memorizzi in un'opportuna matrice e stampi al termine la sottomatrice quadrata, tra quelle precedentemente individuate, la somma dei cui elementi è massima.

- Problema di filtraggio
  - Enumeriamo direttamente le matrici quadrate della dimensione desiderata
- Problema di ottimizzazione
  - Richiesta solo una soluzione
  - Problema di ricerca del massimo

# Operazioni su insiemi

- Insiemi rappresentati da vettori
- Gestione elementi ripetuti
- Vogliamo effettuare operazioni insiemistiche (unione, intersezione, differenza simmetrica)

## Soluzione 1:

- Vettori non ordinati
- Utilizzo di una funzione di *appartenenza*
- Complessità quadratica

## Soluzione 2:

- Vettori ordinati
- Le operazioni sfruttano l'ordinamento per fornire la soluzione in tempo lineare
- Complessità dipendente dall'algoritmo di ordinamento scelto