

Aula V

Dip. di Matematica “G. Castelnuovo”

Univ. di Roma “La Sapienza”

Array e puntatori

Igor Melatti

Slides disponibili (assieme ad altro materiale) in:

<http://www.dsi.uniroma1.it/~melatti/programmazione1.2007.2008.html>

- Scrivere una funzione per determinare se in un array di interi vi sono solo elementi dispari
- Scrivere una procedura che modifichi un array sommando 1 a tutti i suoi elementi
- Scrivere una funzione per determinare se una matrice è ordinata crescentemente
 - fare sia l'ordinamento per righe che quello per colonne
 - supponiamo che la matrice data è $m \times n$
 - allora nell'ordinamento per righe occorre che $A[i][j] < A[h][k]$ tutte le volte che $k = j + 1$ oppure $j = n - 1, k = 0, h = i + 1$
 - invece nell'ordinamento per colonne occorre che $A[i][j] < A[h][k]$ tutte le volte che $h = i + 1$ oppure $i = m - 1, h = 0, k = j + 1$

- Scrivere una funzione iterativa per calcolare la media e la varianza di un array di reali.
 - la media è data dalla formula $\mu(A) = \frac{1}{n} \sum_{i=0}^{n-1} A[i]$
 - la varianza è data dalla formula $\frac{1}{n} \sum_{i=0}^{n-1} (A[i] - \mu(A))^2$
- Scrivere una funzione iterativa per calcolare la media e la varianza di un array di reali in modo che ogni elemento dell'array venga ispezionato solo una volta
 - fatevi prima qualche conticino matematico, sviluppando la somma dei quadrati in 3 somme distinte (ricordate che $(a + b)^2 = a^2 + 2ab + b^2$)
- Scrivere un algoritmo per decidere se in un array con un numero di elementi pari $A[0, \dots, 2N-1]$ gli elementi di indice pari sono ordinati in ordine crescente e quelli di indice dispari in ordine decrescente

- Scrivere una funzione per contare il numero di cambi di segno in un array il cui primo elemento è differente da 0. Un cambio di segno si verifica quando nell'array vi sono due interi di segno opposto contigui o separati solo da 0.

- Scrivere una funzione ricorsiva per calcolare l'intersezione ordinata di due vettori ordinati.
- Scrivere una funzione (iterativa) efficiente che calcoli il numero di coppie (i,j) tali che $A[i]+B[j] > 0$ per due vettori ordinati di interi A e B.
- Dato un array di interi A, scrivere un algoritmo che implementi l'ordinamento per selezione con la variante che ad ogni iterazione si selezionano l'elemento minimo e quello massimo della parte non ordinata si posizionano all'inizio ed alla fine della parte ordinata del vettore.

- Dato un vettore A definiamo una sequenza crescente in A come la sequenza definita induttivamente nel modo seguente:
 - $A[0]$ è il primo elemento della sequenza
 - l'elemento che segue $A[i]$ è l'elemento $A[j]$ tale che $j > i$ e $A[j]$ è il più vicino elemento ad $A[i]$ con la proprietà che $A[j] \geq A[i]$.
 - Ad esempio se A è : 1 3 4 2 4 5 2 3 7 8 6 8 12
 - allora la sequenza è 1 3 4 4 5 7 8 8 12

Scrivere un funzione che dato in input un vettore A di interi lo riorganizzi scrivendo la sequenza crescente definita sopra nella parte sinistra del vettore e gli altri elementi (senza nessun ordine particolare) nella parte destra.

- Dato un array di interi A , scrivere un algoritmo che implementi l'ordinamento per selezione con la variante che ad ogni iterazione si selezionano l'elemento minimo e quello massimo della parte non ordinata si posizionano all'inizio ed alla fine della parte ordinata del vettore
- La parte ordinata del vettore sarà la parte iniziale e la parte finale
- Nella prima si mettono i minimi, nell'altra i massimi
- Nel selection sort c'è la nozione di massimo e minimo k -esimo
 - cioè un elemento è un massimo k -esimo se è più grande di tutti gli elementi tranne k
 - ovvero, se esistono solamente k elementi più grandi di lui
 - Es.: 2 4 6 3 4 1; il massimo 0-esimo è 6, il massimo 1-esimo è 4, il 2-esimo è ancora 4, il 3-esimo è 3 e così via
 - analogamente per il minimo

- Ogni volta che si trovano i minimi e i massimi k -esimi, si scambiano le posizioni
- Supponiamo che il minimo k -esimo sia in posizione i
- Questo vuol dire che il minimo k -esimo è $A[i]$
- Allora in posizione k del vettore ci va $A[i]$
- Cioè $A[k]$ dovrà contenere $A[i]$
- Ma così facendo mi perderei quello c'è adesso in $A[k]$
 - 2 13 34 23 56 0 \rightarrow 0 13 34 23 56 0
- Quindi, si scambiano
 - 2 13 34 23 56 0 \rightarrow 0 13 34 23 56 2

- Per il massimo si fa una cosa analoga, cioè il massimo k -esimo va messo in $A[n - k - 1]$
- Ovvero, il massimo 0-esimo va al posto $A[n - 1]$ (cioè l'ultimo), il massimo 1-esimo va al posto $A[n - 2]$ e così via
- Ovviamente, k deve essere di volta in volta incrementato
- Fino ad arrivare alla metà della lunghezza del vettore
 - $k = 0$, A: 2 13 34 23 56 0 \rightarrow 0 13 34 23 56 2 \rightarrow 0 13 34 23 2 56
 - $k = 1$, A: 0 13 34 23 2 56 \rightarrow 0 2 34 23 13 56 \rightarrow 0 2 13 23 34 56
 - $k = 2$, A: 0 2 13 23 34 56 \rightarrow 0 2 13 23 34 56 \rightarrow 0 2 13 23 34 56

```
int selsort(int A[], int dim)
{
    int i, j = 0, min, max, i_min, i_max;
    while (j < dim/2) {
        min = A[j]; i_min = j;
        max = A[dim - 1 - j]; i_max = dim - 1 - j;
        for (i = j; i < dim - 1 - j; i++) {
            if (A[i] < min) {
                min = A[i]; i_min = i;
            }
            if (A[i] > max) {
                max = A[i]; i_max = i;
            }
        }
        swap(A[j], A[i_min]);          /* attenzione, questo non e' proprio cosi' */
        swap(A[dim - 1 - j], A[i_max]); /* idem */
        j++;
    }
}
```

- ATTENZIONE: così facendo c'è un errore
- Provate a vedere cosa accade nel seguente caso:
- 54 5 34 23 67 35 12
- Provate a capire l'errore (gdb...) e correggerlo