

Counting sort

([Harold H. Seward](#))

Liceo G.B. Brocchi - Bassano del Grappa (VI)
Liceo Scientifico - opzione scienze applicate
Giovanni Mazzocchin

Counting sort

A												
items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

- chiamiamo **A** l'array rappresentato sopra
- imponiamo un vincolo sui valori di A:
 - devono essere compresi tra 0 e 9
- in generale, l'algoritmo che vedremo (**Counting sort**) funziona partendo dall'ipotesi che gli elementi (interi) di A appartengano al range $[0, k]$

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9

- creiamo un array **C** di 10 elementi, tutti inizializzati a 0
- la lunghezza di C dipende dal range di ipotesi: in questo esempio, ciascun elemento di A appartiene all'insieme $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	0	0	0	0	0	0	0	0	0	0
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	0	0	0	0	0	0	0	1	0	0
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	0	0	0	0	0	0	0	1	0	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	0	1	0	0	0	0	0	1	0	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	1	0	0	0	0	0	1	0	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	2	0	0	0	0	0	1	0	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	2	0	0	0	0	1	1	0	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	2	0	0	0	0	2	1	0	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	2	0	0	0	0	2	1	1	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	2	0	0	0	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	3	0	0	0	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	3	0	0	1	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	3	0	1	1	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- facciamo in modo che $C[i]$ contenga il numero di occorrenze di i in A
- è sufficiente un ciclo su A per valorizzare correttamente gli elementi di C

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	3	0	1	1	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A												
items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C										
items	1	3	0	1	1	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	0	1	1	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	1	1	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A												
items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C										
items	1	4	4	5	1	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	5	6	0	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	5	6	6	2	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	5	6	6	8	1	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	5	6	6	8	9	2	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	5	6	6	8	9	11	1
ind.	0	1	2	3	4	5	6	7	8	9

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

A

items	7	9	1	0	1	6	6	8	8	1	4	3
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	1	4	4	5	6	6	8	9	11	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- ora, facciamo in modo che $C[i]$ contenga il numero di occorrenze di tutti gli elementi di A minori o uguali ad i
- se $C[i]$ contiene il numero di occorrenze di i in A, allora $C[0] + \dots + C[i]$ è il numero di occorrenze di tutti gli elementi di A minori o uguali ad i

Counting sort

B

items												
	ind.	0	1	2	3	4	5	6	7	8	9	10

C

items	1	4	4	5	6	6	8	9	11	12
ind.	0	1	2	3	4	5	6	7	8	9

- ora, sfruttiamo C per creare un nuovo array (B) contenente tutti gli elementi di A ordinati in senso ascendente
- **intuizione:** *se hai esattamente tre fratelli più grandi di te, allora tu sei il quarto figlio (o il figlio di indice 3, se numeriamo i figli da 0)*

Counting sort

B

items												
	ind.	0	1	2	3	4	5	6	7	8	9	10

C

items	1	4	4	5	6	6	8	9	11	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- cicliamo sugli indici di A, dall'ultimo al primo, utilizzando la variabile j

Counting sort

B

items					3							
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	4*	6	6	8	9	11	12
ind.	0	1	2	3	4	5	6	7	8	9

- j : 11
- $A[11]$: 3
- $C[3]$: 5
- poniamo 3 in posizione $5-1:4$ di B, poi decrementiamo $C[3]$ di 1

Counting sort

B

items					3	4						
ind.	0	1	2	3	4	5	6	7	8	9	10	11

C

items	1	4	4	4	5*	6	8	9	11	12
ind.	0	1	2	3	4	5	6	7	8	9

- $j: 10$
- $A[10]: 4$
- $C[4]: 6$
- poniamo 4 in posizione $6-1:5$ di B, poi decrementiamo $C[4]$ di 1

Counting sort

B

items				1	3	4						
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	1	3*	4	4	5	6	8	9	11	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 9$
- $A[9]: 1$
- $C[1]: 4$
- poniamo 1 in posizione $4-1:3$ di B, poi decrementiamo $C[1]$ di 1

Counting sort

B

items				1	3	4					8	
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	1	3	4	4	5	6	8	9	10*	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 8$
- $A[8]: 8$
- $C[8]: 11$
- poniamo 8 in posizione $11-1:10$ di B, poi decrementiamo $C[8]$ di 1

Counting sort

B

items				1	3	4				8	8	
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	1	3	4	4	5	6	8	9	9*	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 7$
- $A[7]: 8$
- $C[8]: 10$
- poniamo 8 in posizione $10-1:9$ di B, poi decrementiamo $C[8]$ di 1

Counting sort

B

items				1	3	4		6		8	8	
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	1	3	4	4	5	6	7*	9	9	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- j : 6
- $A[6]$: 6
- $C[6]$: 8
- poniamo 6 in posizione $8-1:7$ di B, poi decrementiamo $C[6]$ di 1

Counting sort

B

items				1	3	4	6	6		8	8	
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	1	3	4	4	5	6	6*	9	9	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 5$
- $A[5]: 6$
- $C[6]: 7$
- poniamo 6 in posizione $7-1:6$ di B, poi decrementiamo $C[6]$ di 1

Counting sort

B

items			1	1	3	4	6	6		8	8	
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	1	2*	4	4	5	6	6	9	9	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 4$
- $A[4]: 1$
- $C[1]: 3$
- poniamo 1 in posizione $3-1:2$ di B, poi decrementiamo $C[1]$ di 1

Counting sort

B

items	0		1	1	3	4	6	6		8	8	
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	0*	2	4	4	5	6	6	9	9	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 3$
- $A[3]: 0$
- $C[0]: 1$
- poniamo 0 in posizione $1-1:0$ di B, poi decrementiamo $C[0]$ di 1

Counting sort

B

items	0	1	1	1	3	4	6	6		8	8	
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	0	1*	4	4	5	6	6	9	9	12
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 2$
- $A[2]: 1$
- $C[1]: 2$
- poniamo 1 in posizione $2-1:1$ di B, poi decrementiamo $C[1]$ di 1

Counting sort

B

items	0	1	1	1	3	4	6	6		8	8	9
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	0	1	4	4	5	6	6	9	9	11*
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 1$
- $A[1]: 9$
- $C[9]: 12$
- poniamo 9 in posizione $12-1:11$ di B, poi decrementiamo $C[9]$ di 1

Counting sort

B

items	0	1	1	1	3	4	6	6	7	8	8	9
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>

C

items	0	1	4	4	5	6	6	8*	9	11
ind.	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

- $j: 0$
- $A[0]: 7$
- $C[7]: 9$
- poniamo 7 in posizione $9-1:8$ di B, poi decrementiamo $C[7]$ di 1

Counting sort

```
counting_sort(A, B, k):  
    for i = 0 to k:  
        C[i] = 0  
  
    for i = 0 to length(A) - 1:  
        item = A[i]  
        C[item] = C[item] + 1  
  
    for i = 1 to k:  
        C[i] = C[i] + C[i - 1]  
  
    for i = length(A) - 1 downto 0:  
        item = A[i]  
        count_lte = C[item]  
        pos_item = count_lte - 1  
        B[pos_item] = item  
        C[item] = C[item] - 1
```