Ordinamento in casi speciali

Dato un array A di n interi in cui solo lg n sono diversi tra loro mentre i rimanenti sono tutti uguali tra loro, descrivere un algoritmo che ordina A in tempo O(n).
Nell'esempio B ha 4 elementi di cui quindi lg 4 = 2 diversi e 2 uguali e A ha 8 elementi di cui lg 8 = 3 diversi e i rimanenti uguali tra loro.

B=

2	5	2	7
0	1	2	3

A=

5	2	2	3	2	2	7	2
0	1	2	3	4	5	6	7

Suggerimento: Utilizzare l'algoritmo 3-PARTITION che divide gli elementi di una sequenza in tre parti: i più piccoli dell'elemento preso come pivot, gli uguali e a seguire i maggiori.

Ordinamento in casi speciali

Soluzione:

1. **Si scorre A alla ricerca del primo duplicato, sia a questo elemento. Poiché solo lg n sono diversi la ricerca termina in al più O(lg n).**
2. **si utilizza la 3-partition utilizzando a come pivot. Alla fine dell'esecuzione i più piccoli dell'elemento ripetuto sono a sinistra, seguiti dai duplicati e poi dai più grandi. Questo comporta un tempo O(n).**
3. **Infine si devono ordinare le due parti dei minori e dei maggiori di a, questi elementi sono al più O(lg n), quindi usando il mergesort possono essere ordinati, separatamente in O(lg n lg lg n)**

A=

5	2	3	3	3	3	1	3
0	1	2	3	4	5	6	7

a = 3

A=

2	1	3	3	3	3	3	5
0	1	2	3	4	5	6	7

A=

1	2	3	3	3	3	3	5
0	1	2	3	4	5	6	7

Il tempo complessivo di esecuzione è in O(n)