

Algoritmi e Strutture di Dati (3^a Ed.)
Errata corrige

Alan Bertossi, Alberto Montresor

Gli autori saranno riconoscenti a chiunque segnali errori presenti nel libro di testo, scrivendo ad `alberto.montresor@unitn.it`.

Errori

- Pag. 22: non considerando la parte intera, la ricorrenza per $T(n)$ non funziona per $n = 1$, che non è pari. Per essere precisi, bisognerebbe introdurre il caso $T(1) = T(0) + d$ (vengono eseguite tutte le operazioni per trovare il mediano e verificare se è il valore cercato, e poi si cerca su un sottovettore vuoto)
- Pag. 47, Es. 2.12, codice `follia()`: nell'assegnamento $i \leftarrow \lfloor n/2 \rfloor$ si modifica l'indice i del for: sostituirlo con `int k ← ⌊n/2⌋`
- Pag. 108: Nel testo dell'Esempio 6.2, l'Esempio 1.2 è in realtà l'Esempio 1.3.
- Pag. 111: Nella figura 6.2, parte alto, lato destro, il valore 33 deve essere sostituito con 37.
- Pag. 124, 125, 125: nel testo, sostituire $\lfloor m/2 \rfloor$ con $\lceil m/2 \rceil$ (3 volte).
- Pag. 125, didascalia figura 6.7: ... con 8 chiavi.
- Pag. 150, in due occasioni, `isEmpty()` deve essere sostituita con `size()`.
- Pag. 152, la funzione `difference()` deve essere sostituita con la seguente:

SET difference(SET A, SET B)

```
SET C ← Set()
foreach s ∈ A do
    if not B.contains(s) then
        C.insert(s)
return C
```

- Pag. 152, sezione 8.3, secondo paragrafo: sostituire la frase “Si usano questa volta tre variabili per scandire A, B e C” con “Si usano questa volta due variabili per scandire A e B.
- Pag. 152, “Esempio 2.10 è in realtà “Esempio 2.11”
- Pag. 153, la funzione `interesection()` deve essere sostituita con la seguente:

LIST intersection(LIST A, LIST B)

```
LIST C ← Set()
POS p ← A.head()
POS q ← B.head()
while not A.finished(p) and not B.finished(q) do
    if A.read(p) = B.read(q) then
        C.insert(C.tail(), A.read(p))
        p ← A.next(p)
        q ← B.next(q)
    else if A.read(p) < B.read(q) then
        p ← A.next(p)
    else
        q ← B.next(q)
return C
```

- Pag. 156, sostituire “tre variabili” con “due variabili”; la funzione union() deve essere sostituita con la seguente:

```

LIST union(LIST A, LIST B)
LIST C ← Set()
POS p ← A.head()
POS q ← B.head()
while not A.finished(p) and not B.finished(q) do
    if A.read(p) = B.read(q) then
        C.insert(C.tail(), A.read(p))
        p ← A.next(p)
        q ← B.next(q)
    else if A.read(p) < B.read(q) then
        C.insert(C.tail(), A.read(p))
        p ← A.next(p)
    else
        C.insert(C.tail(), B.read(q))
        q ← B.next(q)
while not A.finished(p) do
    C.insert(C.tail(), A.read(p))
    p ← A.next(p)
while not B.finished(q) do
    C.insert(C.tail(), B.read(q), r)
    q ← B.next(q)
return C

```

- Pag. 157, la funzione difference() deve essere sostituita con la seguente:

```

SET difference(LIST A, LIST B)
LIST C ← Set()
POS p ← A.head()
POS q ← B.head()
while not A.finished(p) and not B.finished(q) do
    if A.read(p) < B.read(q) then
        C.insert(C.tail(), A.read(p))
        p ← A.next(p)
    else
        if A.read(p) = B.read(q) then
            p ← A.next(p)
            q ← B.next(q)
while not A.finished(p) do
    C.insert(C.tail(), A.read(p))
    p ← A.next(p)
return C

```

- Pag. 157, esercizio 8.6: Inoltre la complessità è $O(\max\{m_a, n_a\} + \max\{m_b, n_b\})$ nel caso della union(), $O(\max\{m_a, n_a\})$ nel caso della difference().
- Pag. 164, Soluzione di Holmes: Il ciclo (2) è “A,G,H,B,A” e non “A,G,A,B,A”
- Pag. 176, algoritmo scc(): la variabile S, di tipo STACK, non è dichiarata.
- Pag. 177, algoritmo ts-dfs(): la chiamata ricorsiva si effettua sul nodo v, non u: ts-dfs(G, v, visitato, S)

- Pag. 182, es. 9.4: sostituire il codice della soluzione con il seguente

```

boolean bipartito(GRAPH G, NODE r)
  QUEUE S ← Queue()
  S.enqueue(r)
  int[] color ← new int[1...G.n]
  foreach u ∈ G.V() − {r} do color[u] ← senzacolore
  color[r] ← rosso
  while not S.isEmpty() do
    NODE u ← S.dequeue()
    foreach v ∈ G.adj(u) do
      if color[v] = senzacolore then
        color[v] ← 1 − color[u]
        S.enqueue(v)
      else
        if color[v] ≠ color[u] then
          return false
  return true

```

- Pag. 183, es. 9.7: sostituire le righe:

```

int j ← ordine[i]
partenza[j] ← partenza[j] + durata[j]

```

con la riga:

```

partenza[ordine[i]] ← partenza[ordine[i − 1]] + durata[ordine[i]]

```

- Pag. 196, Fig. 10.5, l'ultima sottofigura in basso a destra deve essere etichettata (*g*), non *d*
- Pag. 205, es. 10.3, codice `merge()`: la procedura deve restituire un vettore di tipo `int[]`.
- Pag. 216, sezione 11.5: “conserva un costo di $O(\log n)$ per l'operazione `deleteMin()`, ma richiede un costo ammortizzato di $O(1)$ per le operazioni `insert()` e `decrease()`”.
- Pag. 235, equazione di ricorrenza algoritmo di Strassen: $7T(n/2) - cn^2$ va sostituito con $7T(n/2) + cn^2$.
- Pag. 240, Es. 12.2. Nel codice `maxsum()` sostituire `for k ← i to j do` con `for k ← i to n do`
- Pag. 241, Es. 12.2. Nel codice `maxsumRic()` sono presenti i seguenti errori:
 1. i parametri *A*, *i* e *j* vanno in corsivo e non in grassetto
 2. sostituire $\max'_d \leftarrow 0$ con $\max'_s \leftarrow 0$
 3. sostituire $\max'_s 0$ con $\max'_d \leftarrow 0$
 4. l'istruzione **return** va fuori dal **for**
 5. nei parametri della **return** va sostituito $\max'_s, \max'_d$ con $\max'_s + \max'_d$
- Pag. 261, definizione ricorsiva di $D[i, c]$ (e conseguentemente, algoritmo `zaino()`): l'ordine delle condizioni per $c < 0$ e per $i = 0 \vee c = 0$ va scambiato.

$$D[i, c] = \begin{cases} -\infty & \text{se } c < 0 \\ 0 & \text{se } i = 0 \vee c = 0 \\ \max\{D[i-1, c], D[i-1, c-v_i] + p[i]\} & \text{altrimenti} \end{cases}$$

- Pag. 272, Essendo un vettore di valori compresi nel range di numeri reali $[0, 1]$, il vettore *x* di `ZAINO()` deve essere dichiarato come `real[]`. I vettori *p*, *v* e la capacità *C* possono essere valori reali, diversamente dallo Zaino 0-1.

- Pag. 281, Esercizio 14.7: rimuovere “di lunghezza k ” e “con $k_i \in [0, k]$ ” dal testo dell’esercizio, in quanto non usato nella soluzione.
- Pag. 318, algoritmo cavallo(). Riga 3: **for integer** $i \leftarrow 1$ **to** n **do** deve essere sostituito con **for integer** $i \leftarrow 1$ **to** 8 **do**.

Sviste minori

- Pag. VII, Titolo della sezione 9.5.8: Applicazione schema DFS: Ordinamento topologico.
- Pag. 81, funzione hanoi-iterativa(), manca la parentesi di chiusura.
- Pag. 91, il metodo leftmostchild() dovrebbe essere scritto leftmostChild()
- Pag. 127: ... B+-albero di figura 6.7.
- Pag. 176: “La procedura topSort(), coadiuvata da ts-dfs()” (non topsortdfs())
- Pag. 198, pseudocodice merge(): i parametri x e y vanno scritti in corsivo.
- Pag. 325, seconda riga di testo: per chiarezza, “non sembra essere un problema difficile” va sostituito con “non è un problema difficile”.
- Pag. 351, quart’ultima riga: per chiarezza, “...ma lui lo ha rifiutato, in precedenza, ...” va sostituito con “...ma lui lo ha rifiutato. In precedenza, ...”.
- Pag. 374: l’ultimo capoverso va indentato.
- Pag. 374, quart’ultima riga: sostituire “Anche in questo caso” con “In questo caso”.
- Pag. 393, bibliografia 14) greedy: togliere l’articolo di J.M. Moore (1968).