

Algoritmi e Strutture Dati – Parte A – 14/01/2020

Esercizio A1 – Punti ≥ 9

Si calcoli la complessità computazionale nei casi pessimo, ottimo, medio della seguente procedura.

```

fun(int[] A, int i, int j)
  if i < j then
    for k = i to j do
      InsertionSort(A, i, j) % InsertionSort between i and j
      int s = random(i, j) % Generate a random number between i and j, included, in O(1)
      int Δ = random(1, 100) % Generate a random number between 1 and 100, included, in O(1)
      A[s] = A[s] + Δ
    int quarter = ⌊(j - i + 1)/4⌋
    fun(A, i, i + quarter · 1 - 1)
    fun(A, i + quarter · 1, i + quarter · 2 - 1)
    fun(A, i + quarter · 2, i + quarter · 3 - 1)
    fun(A, i + quarter · 3, j)

```

Esercizio A2 – Punti ≥ 9

Sia A un vettore contenente n valori interi appartenenti all'intervallo $[1 \dots 9]$. Si parte dalla posizione 1 del vettore. Quando ci si trova in posizione i , è possibile fare un passo e andare:

- nella posizione $i - 1$
- nella posizione $i + 1$
- oppure in qualunque casella j tale per cui $A[i] = A[j]$

Scrivere un algoritmo

```
int minSteps(int[] A, int n)
```

che restituisca il numero minimo di passi per andare dalla posizione 1 alla posizione n .

Discutere informalmente la correttezza dell'algoritmo e calcolare la sua complessità computazionale.

Esempio: nel vettore $A = [1, 9, 3, 9]$, è possibile andare dalla posizione $A[1]$ alla posizione $A[2]$; da lì, è possibile andare direttamente in $A[4]$ perché $A[2]$ e $A[4]$ hanno lo stesso valore. È quindi possibile andare da 1 a 4 in due passi, e l'algoritmo deve restituire 2.

Esercizio A3 – Punti ≥ 12

Sia A un vettore ordinato contenente n interi distinti e sia B un vettore ordinato contenente $n + 1$ interi distinti. I due vettori contengono gli stessi valori, tranne un valore che è stato aggiunto in B . Scrivere un algoritmo

```
int intruder(int[] A, int[] B, int n)
```

che restituisca il valore aggiunto.

Spiegare il funzionamento e discutere la complessità computazionale dell'algoritmo proposto. *Soluzioni in tempo lineare o superiore non verranno considerate.*

Esempio:

```
A = [ 2, 4, 6, 8, 10, 12 ]
```

```
B = [ 2, 4, 6, 8, 9, 10, 12 ]
```

L'algoritmo deve restituire il valore 9.

