Cognome: # Matricola: Riga: Col:

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

Esercizio -1 Iscriversi allo scritto entro la scadenza. In caso di inadempienza, -1 al voto finale.

Esercizio 0 Scrivere correttamente nome, cognome, numero di matricola, riga e colonna su tutti i fogli consegnati. Consegnare foglio A4 e foglio protocollo di bella. In caso di inadempienza, -1 al voto finale.

Esercizio A1 – Punti > 9

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

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. E' 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.

Discutere informalmente la correttezza dell'algoritmo e calcolare la sua complessità computazionale. Algoritmi di complessità $\Theta(n)$ non verranno considerati in quanto banali.

Esempio:

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

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

L'algoritmo deve restituire il valore 9.

Cognome: # Matricola: Riga: Col:

Algoritmi e Strutture Dati - Parte B - 14/01/2020

Esercizio -1 Iscriversi allo scritto entro la scadenza. In caso di inadempienza, -1 al voto finale.

Esercizio 0 Scrivere correttamente nome, cognome, numero di matricola, riga e colonna su tutti i fogli consegnati. Consegnare foglio A4 e foglio protocollo di bella. In caso di inadempienza, -1 al voto finale.

Esercizio B1 – Punti > 8

Sia X,Y due vettori che rappresentano n coppie di interi; gli elementi della i-esima coppia sono denotati X[i],Y[i] e vale la regola X[i] < Y[i]. Una catena di coppie è una sequenza di indici i_1,\ldots,i_m , tale per cui $Y[i_k] \leq X[i_{k+1}]$, per ogni $1 \leq k < m$.

Scrivere un algoritmo

$$int longestChain(int[] X, int[] Y, int n)$$

che restituisca la lunghezza della più lunga catena di coppie presente nei vettori di input, misurata in numero di coppie.

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

Esercizio B2 – Punti > 10

Dato un vettore A contenente n interi, scrivere un algoritmo

int maxSumEven(int[]
$$A$$
, int n)

che restituisca il valore del sottovettore di lunghezza pari $A[i\dots j]$ di somma massimale, ovvero il sottovettore la cui somma degli elementi $\sum_{k=i}^{j} A[k]$ sia più grande o uguale alla somma degli elementi di qualunque altro sottovettore di lunghezza pari.

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

Esempio:

- Input: A = [1, -2, 3], output 1, in quanto il sottovettore massimale di lunghezza pari è [-2, 3]
- Input: A = [8, 9, -8, 9, 10], output 20, in quanto il sottovettore massimale di lunghezza pari è [9, -8, 9, 10]
- Input: A = [-2, 1, -2, 1, -2], output 0, in quanto tutti i settovettori di lunghezza pari hanno somma negativa, tranne quello di lunghezza zero che ha somma 0.

Esercizio B3 – Punti > 12

Sia M una matrice $n \times m$ contenente valori 0/1. Un percorso di valori 1 è una sequenza contigua di celle adiacenti (orizzontale-verticale, non diagonale). La sua lunghezza è il numero di celle che la compongono.

Scrivere un algoritmo

int longestPath(int[][]
$$M$$
, int n , int m)

che restituisca la lunghezza del percorso più lungo che inizia nella cella (1, 1).

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

La figura qui sotto illustra un percorso di valori 1 di lunghezza 27, che è massimale per questa matrice 3×10 .

