

Algoritmi e Strutture Dati – Parte A – 21/08/2018

A1 – Complessità – Punti ≥ 6

Trovare i limiti superiore e inferiori più stretti possibili per la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 3T(\lfloor n/3 \rfloor) + 4T(\lfloor n/4 \rfloor) + \\ \quad 12T(\lfloor n/6 \rfloor) + n^2 & n > 6 \\ 1 & n \leq 6 \end{cases}$$

A2 – Minecraft – Punti ≥ 12

In Minecraft, una carta geografica altimetrica è rappresentata da una matrice $n \times n$ di celle quadrate, ognuna delle quali riporta un'altitudine intera misurata in blocchi. Un valore minore o uguale a zero corrisponde ad un mare, mentre un valore positivo corrisponde ad un'isola.

Scrivere un algoritmo che prenda in input una matrice A di interi e la sua dimensione positiva n , e restituisca l'altezza media dell'isola più elevata.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

Ad esempio, nella matrice seguente ci sono tre isole, una 2×3 in alto a sinistra con altezza media 1.5, una 4×1 a destra con altezza media 1, una 1×2 in basso a sinistra con altezza media 2. L'isola con altezza media più alta è quella in basso a sinistra, e quindi l'algoritmo deve restituire 2.

1	1	1	0	1
2	2	2	-1	1
0	-1	0	-2	1
0	0	-1	-2	1
2	2	-1	0	0

A3 – Vettori traslati – Punti ≥ 12

Sia A un vettore ordinato contenente n interi positivi distinti, che è stato traslato ("shiftato") di $k < n$ posizioni a destra. Ad esempio, il vettore $[35, 42, 1, 7, 15, 18, 28, 30]$ è un vettore ordinato che è stato shiftato $k = 2$ posizioni a destra, mentre il vettore $[15, 18, 28, 30, 35, 42, 1, 7]$ è un vettore ordinato che è stato shiftato $k = 6$ posizioni a destra.

Scrivere un algoritmo

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

che prende un vettore ordinato traslato A come input e restituisce il valore di k corrispondente. L'algoritmo deve operare in tempo $O(\log n)$ (non saranno considerate soluzioni in tempo lineare o superiore).

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

Algoritmi e Strutture Dati – Parte B – 21/08/2018

B1 – Hateville Clubs – Punti ≥ 8

Ad Hateville, città famosa per le sue faide di vicinato, abitano un insieme P di persone, di dimensione $|P|$. Esiste inoltre un insieme C di club, di dimensione $|C|$. Ogni persona $i \in P$ può appartenere ad un sottoinsieme $C_p[i] \subseteq C$ dei club presenti. Infine, esiste un insieme Z di partiti politici, con dimensione pari a $|Z|$. Ogni persona $i \in P$ appartiene esattamente ad un partito politico, memorizzato in $Z_p[i]$.

È arrivato il momento di nominare il nuovo consiglio dei club. In tale consesso, ci dovrà essere esattamente un membro per ogni club. Ogni membro può rappresentare uno solo dei club a cui è iscritto. Per evitare diatribe politiche, è importante che nessuno dei partiti abbia una maggioranza assoluta di persone in consiglio (ovvero, che abbia più di $|C|/2$ membri).

I cittadini di Hateville si sono rivolti a voi per sbrogliare la matassa. Dati gli insiemi P , C , Z , gli assegnamenti persona-club C_p e l'assegnamento persona-partito Z_p , descrivere un algoritmo che restituisca un sottoinsieme $X \subseteq P$ composto da $|C|$ persone, che rispetta le regole di cui sopra, se tale insieme esiste.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

B2 – Più che doppio – Punti ≥ 10

Un vettore di numeri interi positivi è detto *più che doppio* se dati due suoi elementi successivi $V[i]$ e $V[i+1]$, si ha che $V[i+1] > 2V[i]$. Un vettore con zero o un elemento è banalmente *più che doppio*.

Dato un vettore V e la sua dimensione n , scrivere un algoritmo che restituisca la lunghezza della più lunga sottosequenza *più che doppia* presente nel vettore. Ricordiamo che in una sottosequenza, i valori selezionati non devono essere necessariamente consecutivi.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

Esempi:

- Il vettore $V = [1, 3, 7, 15, 31, 63]$ è *più che doppio*. Con questo input, la risposta è 6.
- Il vettore $V = [5, 1, 4, 7, 2, 9]$ non è *più che doppio*; tuttavia, la sottosequenza $[1, 4, 9]$ lo è, ed è la più lunga possibile. Con questo input, la risposta è 3.

B3 – n -colorazione con gap – Punti ≥ 12

Sia G un grafo *non orientato* e *connesso* contenente n nodi. Una n -colorazione con gap del grafo è un assegna-

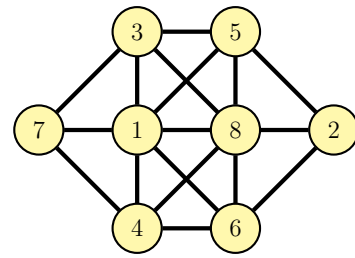
mento dei numeri compresi fra 1 ed n ai nodi del grafo che rispetta le seguenti regole:

- Tutti i numeri in $\{1, \dots, n\}$ sono usati esattamente una volta
- Dati due nodi adiacenti u, v (i.e., $(u, v) \in E$) e detti $color[u], color[v]$ i loro colori, allora $|color[u] - color[v]| \geq 2$. In altre parole, non è possibile assegnare due colori uguali o consecutivi a due nodi adiacenti.

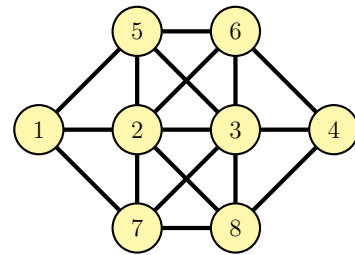
Scrivere un algoritmo che prenda in input un grafo G e restituisca **true** se e solo se esiste una n -colorazione con gap G .

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

L'esempio successivo mostra una 8-colorazione di un grafo con 8 nodi. Le regole sono rispettate: tutti i colori sono utilizzati e non ci sono nodi adiacenti con colori consecutivi o uguali.



L'esempio successivo mostra un tentativo fallito di creare una 8-colorazione con gap, in quanto (tanti) numeri consecutivi sono assegnati a nodi adiacenti.



L'esempio successivo mostra un tentativo fallito di creare una 8-colorazione con gap, in quanto alcuni numeri sono assenti.

