Cognome: **Nome:** **# Matricola:** **Riga:** **Col:**

Algoritmi e Strutture Dati – 17/12/15

Esercizio 1 – Punti ≥ 4 (Parte B)

Si consideri il problema di colorare un grafo non orientato $G = (V, E)$, ovvero di assegnare un colore (rappresentato da un intero da 1 ad n) ad ogni nodo in modo tale che due nodi adiacenti non abbiano lo stesso colore. Un vostro amico propone il seguente algoritmo, descritto informalmente: esamina i nodi in qualche ordine, assegnando ad ogni nodo u il primo colore (quello con valore minore) fra quelli non utilizzati dai nodi adiacenti di u.
Il vostro amico afferma che questo algoritmo utilizza il minimo numero possibile di colori, in quanto utilizza al più $d + 1$ colori, dove d è il grado massimo del grafo (in altre parole: non è possibile colorare il grafo con meno colori). Dimostrare che l'affermazione del vostro amico è corretta, oppure produrre un controesempio.

Esercizio 2 – Punti ≥ 8 (Parte B)

Si consideri una sequenza di interi V. La cancellazione da V di un certo numero di elementi, mantenendo l'ordine, determina una sottosequenza. Una k-sottosequenza di V è una sottosequenza di V in cui compaiono al più k elementi consecutivi di V. Il valore di una sottosequenza è dato dalla somma dei suoi elementi.
Dato un vettore V contenente n interi ed un intero k, con $k \leq n$, scrivere un algoritmo che restituisca il valore della k-sottosequenza massimale, ovvero quella che ha valore massimo fra tutte le k-sottosequenze.

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

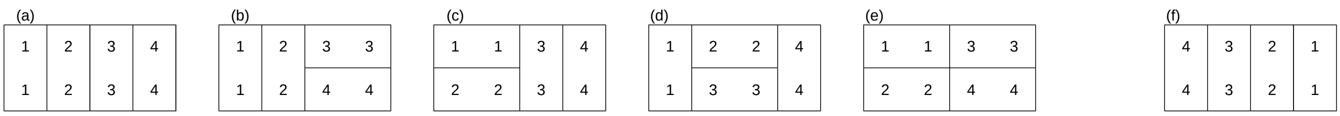
Ad esempio, per $V = \{9, 1, 9\}$ e $k = 2$, l'output è 18, ottenuto dalla 2-sottosequenza massimale $\{9, 9\}$.
Ad esempio, per $V = \{1, 7, 8, 9, 1, 10, 6, 8, 8\}$ e $k = 2$, l'output è 44, ottenuto dalla 2-sottosequenza massimale $\{1, 8, 9, 10, 8, 8\}$.
Ad esempio, per $V = \{1, 7, 8, 9, 1, 10, 6, 8, 8\}$ e $k = 3$, l'output è 50, ottenuto dalla 3-sottosequenza massimale $\{7, 8, 9, 10, 8, 8\}$.

Esercizio 3 – Punti ≥ 10 (Parte B)

Il gioco del domino contiene tessere di dimensione 2×1. Si considerino le disposizioni di n tessere all'interno di un rettangolo $2 \times n$. Scrivere un algoritmo che conti tutte le disposizioni possibili.

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

I casi (a)-(e) della figura rappresentano i cinque modi possibili con cui è possibile riempire un rettangolo 2×4.



Esercizio 4 – Punti ≥ 10 (Parte B)

Siano dati in input una rete di flusso $G = (V, E, s, p, c)$, rappresentata da una matrice di capacità positive **int**[ ][ ] c, di dimensione $n \times n$, tale per cui $(u, v) \in E \Leftrightarrow c[u, v] > 0$, e da due interi s, p che rappresentano gli indici dei nodi sorgente e pozzo. Sia inoltre dato in input un flusso massimo **int**[ ][ ] f, già calcolato, per la rete di flusso definita da c, s, p.

Sia data una coppia di indici di nodi u, v tale per cui $c[u, v] > 0$. Scrivere un algoritmo in pseudocodice che riduce la capacità $c[u, v]$ di una unità e calcola il flusso massimo definito sulla nuova rete di flusso così ottenuta.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.
Alcune note: a differenza dei normali esercizi sul flusso, è necessario scrivere dello pseudocodice. Zero punti per soluzioni che semplicemente modificano il valore della capacità e lanciano uno degli algoritmi visti a lezione con complessità elevata. Per comodità, riporto la firma dell'algoritmo che dovete scrivere:

reduceFlow(**int**[ ][ ] c, **int** n, **int** s, **int** p, **int**[ ][ ] f, **int** u, **int** v)