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

Algoritmi e Strutture Dati - Seconda provetta
31/05/12

Esercizio 0 Scrivere correttamente nome, cognome, numero di matricola, riga e colonna.

Esercizio 1 – Punti ≥ 6 (Parte B)

Si consideri la rete di flusso di Figura 1. Siano $x[1] \ldots x[6]$ le 6 cifre del vostro numero di matricola, da sinistra (più significativa) a destra (meno significativa). Ovvero, se il vostro numero di matricola è 123456, $x[1] = 1$, $x[2] = 2$, $x[3] = 3$, etc. Identificare il valore totale del flusso massimo. Nel foglio di consegna, è obbligatorio riportare solo il valore totale del flusso massimo, non è necessario riportare i valori sugli archi o illustrare il procedimento.

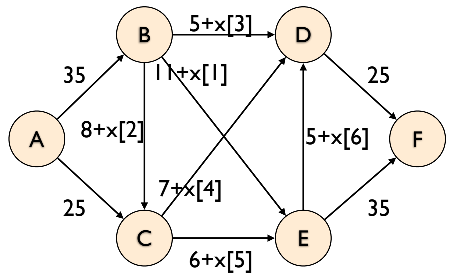

Figura 1: Problema di flusso

Esercizio 2 – Punti ≥ 6 (Parte B)

Una sequenza di interi $X = x_1 x_2 \ldots x_m$ si definisce ZIG-ZAG se, per $1 \leq i < m$,

$$\begin{array}{ll} x_i < x_{i+1} & \text{se } i \text{ è dispari} \\ x_i > x_{i+1} & \text{se } i \text{ è pari} \end{array}$$

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

Ad esempio $X = (3, 8, 1, 5, 2)$ è una sequenza ZIG-ZAG, mentre $X = (3, 8, 10, 5, 2)$ non lo è. Descrivere ed analizzare un algoritmo che data una sequenza $Y = y_1 y_2 \ldots y_n$ restituisca la lunghezza della più lunga sottosequenza ZIG-ZAG di Y. Ad esempio, se $Y = (3, 4, 8, 5, 6, 2)$ allora la lunghezza massima è 5 (ossia la sottosequenza è $3, 8, 5, 6, 2$ o anche $4, 8, 5, 6, 2$).
Esiste un algoritmo $O(n)$ basato su tecnica greedy.

Esercizio 3 – Punti ≥ 10 (Parte B)

Supponete di avere n interi positivi, distinti, memorizzati in un vettore $B[1 \ldots n]$, che rappresentano tagli di banconote, e un valore intero positivo T, che rappresenta un resto da dare. Scrivere un algoritmo che *conta* il numero totale di modi diversi per restituire questo resto, sommando un numero qualsiasi di banconote. Discutere correttezza e complessità.

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

Ad esempio: Supponiamo di avere banconote da $1, 2, 5$ euro e di dover dare un resto di 4 euro. I modi diversi di dare questo resto sono:

$$\{1 + 1 + 1 + 1\}, \{1 + 1 + 2\}, \{2 + 2\}$$

Notate che $\{2 + 1 + 1\}$ e $\{1 + 2 + 1\}$ non contano come modi diversi perchè sono permutazioni della seconda. Notare che è possibile rispondere 0; per esempio se ho una banconota da 5 e devo dare resto 7. È possibile risolvere il problema tramite backtrack, ma la programmazione dinamica è applicabile in questo caso.

Esercizio 4 – Punti ≥ 10 (Parte A)

Siano $X[1 \ldots n]$ e $Y[1 \ldots n]$ due vettori, ciascuno contenente n interi già ordinati. Scrivere un algoritmo che trovi i valori mediani dei $2n$ elementi dei vettori X e Y presi insieme. Usiamo il plurale perchè essendo $2n$ pari, è possibile definire *due* valori mediani.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale. Tramite divide-et-impera, è possibile trovare la mediana in tempo $O(\log n)$. Algoritmi meno efficienti verranno considerati ma non valutati con punteggio pieno.