Algoritmica Avanzata – CdL Magistrale in Ingegneria Informatica

Compito, 19/6/2012 (Durata: 2h)

Nome, Cognome, Matricola: ______________________________

Risoluzione di problemi

Si forniscano soluzioni esaurienti e rigorose ai tre problemi seguenti. Gli algoritmi vanno codificati utilizzando lo **pseudocodice** usato in classe. **Attenzione:** Risposte immotivate e prive di prova di correttezza non saranno considerate.

Esercizio 1 [12 punti] Dato un grafo **orientato senza self-loops** $G = (V, E)$, con $V \subseteq \boldsymbol{N}$ e $E \subseteq V \times V - \{(v, v) : v \in V\}$, si vuole determinare un sottoinsieme $E^\star \subseteq E$ di massima cardinalità per cui il sottografo indotto $G' = (V, E^\star)$ risulti essere aciclico. Si fornisca un algoritmo di 2-approssimazione per il problema. (*Suggerimento:* Si osservi che i vertici sono numeri. Si partizionino gli archi di E in due insiemi in modo tale che un ciclo non possa avere tutti gli archi appartenenti allo stesso insieme ...).

Esercizio 2 [10 punti] Si determini l'inverso moltiplicativo di 17 in $\boldsymbol{Z}_{43}^\star$ attraverso l'applicazione dell'algoritmo EXTENDED_EUCLID.

Esercizio 3 [11 punti] Data una costante $c > 1$, si determini un limite superiore $m_c(n)$ al numero di estrazioni con reimbussolamento da effettuare da un'urna contenente i primi n numeri naturali che garantisca che il numero medio di numeri distinti estratti sia almeno n/c. Per ogni valore di c fissato, il limite superiore deve essere $m_c(n) = O(n)$.

Algoritmica Avanzata – CdL Magistrale in Ingegneria Informatica

Compito, 5/7/2012 (Durata: 2h)

Nome, Cognome, Matricola: ______________________________

Risoluzione di problemi

Si forniscano soluzioni esaurienti e rigorose ai tre problemi seguenti. Gli algoritmi vanno codificati utilizzando lo **pseudocodice** usato in classe. **Attenzione:** Risposte immotivate e prive di prova di correttezza non saranno considerate.

Esercizio 1 [11 punti] IL problema del *maximum 4-2-coloring* è definito come segue. Dato un grafo non orientato $G = (V(G), E(G))$, si colori ogni vertice $v \in V$ con uno di 4 colori, $c_{\text{opt}}(v) \in \{1, 2, 3, 4\}$ in modo tale da massimizzare il numero di archi bicromatici

$$B^G_{c_{\text{opt}}} = |\{e = \{u, v\} \in E(G) : c_{\text{opt}}(u) \neq c_{\text{opt}}(v)\}|.$$

Si fornisca un algoritmo di approssimazione randomizzato $\mathrm{A}(G)$ che ritorni una colorazione c_A per cui $\rho = B^G_{c_{\text{opt}}}/E[B^G_{c_A}] \leq 4/3$. (*Suggerimento:* Si colorino i nodi a caso...).

Esercizio 2 [11 punti] Si determini una soluzione in $\boldsymbol{Z}_{91}$ al seguente sistema di congruenze:

$$\begin{cases} x & \equiv & 4 \bmod 7 \\ x & \equiv & 12 \bmod 13 \end{cases}$$

Si descriva il processo utilizzato per giungere alla soluzione e si argomenti che la soluzione ritornata è l'unica possibile.

Esercizio 3 [11 punti] Sia dato un generatore di bit casuali R_BIT(), che ritorna 0 oppure 1 entrambi con probabilità 1/2, e un valore $n > 1$ arbitrario (non necessariamente una potenza di due). Si vuole progettare un generatore R_NUM(n) che, utilizzando R_BIT() come subroutine, ritorni un valore casuale intero X tra 0 e $n - 1$ tale che $\forall i, 0 \leq i < n : \Pr(X = i) = 1/n$. Si fornisca lo pseudocodice di R_NUM(n), se ne provi la correttezza e si valuti un limite superiore al numero medio di chiamate a R_BIT() effettuate da R_NUM(n).

Algoritmica Avanzata – CdL Magistrale in Ingegneria Informatica

Compito, 13/9/2012 (Durata: 2h)

Nome, Cognome, Matricola: Alberto Lazzarin, 1020257

Risoluzione di problemi

Si forniscano soluzioni esaurienti e rigorose ai tre problemi seguenti. Gli algoritmi vanno codificati utilizzando lo **pseudocodice** usato in classe. **Attenzione:** Risposte immotivate e prive di prova di correttezza non saranno considerate.

Esercizio 1 [11 punti] In un grafo non diretto $G = (V, E)$ un *node-cut* è una partizone $(W, V - W)$ di V. La taglia di un node-cut è definita come $s(W, V - W) = |\{e = \{u, v\} \in E : (u \in W) \wedge (v \in V - W)\}|$. Il problema di determinare un node-cut di taglia massima è NP-Hard. Si considerila seguente strategia greedy per il problema: si parta da un arbitrario node-cut (ad esempio, $(V, \emptyset)$). Finchè esiste un nodo che, spostato da un sottoinsieme all'altro del node-cut corrente, ne aumenta la taglia, allora lo si sposti, si aggiorni il node-cut e si ripeta. Altrimenti, si restituisca in uscita il node-cut corrente. Si noti che questo algoritmo termina, in quanto la taglia del node-cut corrente aumenta sempre a ogni iterazione.

Si dimostri che la strategia greedy illustrata fornisce una 2-approssimazione al problema. (*Suggerimento:* Si considerino i vicini di ogni nodo di V nel node-cut finale... Quanti devono trovarsi dal lato opposto?).

Esercizio 2 [11 punti] Si determini una soluzione in $\boldsymbol{Z}_{91}$ al seguente sistema di congruenze:

$$\begin{cases} x \equiv 2 \bmod 7 \\ x \equiv 5 \bmod 13 \end{cases}$$

Si descriva il processo utilizzato per giungere alla soluzione e si argomenti che la soluzione ritornata è l'unica possibile.

Esercizio 3 [11 punti] Si consideri lo stesso problema dell'Esercizio 1. Un semplicissimo algoritmo randomizzato per il problema è il seguente. Inizialmente, $W = \emptyset$. Per ogni $v \in V$, si ponga v in W con probabilità $1/2$. Alla fine, si ritorni il node-cut $(W, V - W)$. Sia s_{opt} la taglia massima di un node-cut. Si dimostri che $E[s(W, V - W)] \geq s_{opt}/2$.