Algoritmi e Strutture Dati 2

Anno Accademico: 2021-2022

Secondo Homework

Docente: Francesco Pasquale

14 gennaio 2022

Consegna: 21 gennaio 2022 ore 19:00

Modalità di consegna. Ogni studente deve consegnare un unico file compresso (possibilmente in formato `zip`) contenente un file con l'elaborato (possibilmente in formato `pdf`) e i sorgenti dei programmi. Il file va inviato per posta elettronica a pasquale@mat.uniroma2.it entro le ore `19:00` di venerdì 21 gennaio 2022. Inserire nella mail nome, cognome e numero di matricola.

Collaborazioni. È consentita e incoraggiata la collaborazione fra gli studenti al fine di risolvere gli esercizi. Tuttavia ogni studente deve poi scrivere il proprio elaborato individualmente e in modo autonomo.

Consigli. Scrivere le soluzioni in modo chiaro e conciso. Una soluzione corretta ma spiegata in modo poco chiaro o eccessivamente prolisso non prende il punteggio massimo. Viceversa, anche una soluzione non corretta può prendere qualche punto se presentata in modo ragionato.

Esercizio 1. Dato un grafo $G = (V, E)$ e un sottoinsieme di nodi $W \subseteq V$, indichiamo con $G[W] = (W, F)$ il *sottografo indotto* da W, ossia il grafo formato dai nodi di W e da tutti gli archi di E che hanno entrambi gli estremi in W, $F = E \cap \binom{W}{2}$.

Progettare e analizzare un algoritmo che prenda in input un grafo $G = (V, E)$ con $|V| = n$ nodi e $|E| = m$ archi e restituisca in output un sottoinsieme di nodi $W \subseteq V$ tale che il sottografo indotto $G[W]$ contenga almeno $\frac{mk(k-1)}{n(n-1)}$ archi. L'algoritmo può essere o deterministico con *running time* polinomiale, oppure probabilistico con *running time* atteso polinomiale.

Esercizio 2. Un *k-labeling* di un grafo $G = (V, E)$ è una funzione $f : V \to \{1, 2, \ldots, k\}$ che associa ad ogni vertice una "etichetta" da un insieme di k etichette (indichiamo le etichette con i numeri interi fra 1 e k). Un grafo si dice *k-colorabile* se esiste un k-labeling f tale che $f(u) \neq f(v)$ per ogni arco $\{u, v\} \in E$. Una terna di nodi $\{u, v, w\} \subseteq V$ è un *triangolo* se E contiene tutti e tre gli archi $\{u, v\}, \{u, w\}, \{v, w\}$. Dato un labeling f, un triangolo $\{u, v, w\}$ si dice *monocromatico* se tutti e tre i nodi hanno la stessa etichetta $f(u) = f(v) = f(w)$.

1. Dimostrare che se un grafo G è 3-colorabile allora esiste un 2-labeling di G senza triangoli monocromatici.
2. Progettare un algoritmo che prenda in input un grafo G e restituisca un 2-labeling di G tale che se G è 3-colorabile allora il 2-labeling restituito dall'algoritmo non contenga triangoli monocromatici. Dare una spiegazione intuitiva e sintetica del perché il vostro algoritmo dovrebbe essere corretto e di quale dovrebbe essere il suo *running time*.
3. (Facoltativo) Analizzare rigorosamente correttezza e *running time* del vostro algoritmo.

Esercizio 3. Si consideri una pedina che si muove su un anello formato da $n+1$ nodi $\{0, 1, \ldots, n\}$. La pedina parte dal nodo 0 e ad ogni istante si muove di una posizione in senso orario o antiorario con la stessa probabilità (ossia, da un nodo $u \in \{0, 1, \ldots, n\}$ si muove sul nodo $u + 1 \pmod{n+1}$ con probabilità $1/2$ e sul nodo $u - 1 \pmod{n+1}$ con probabilità $1/2$). La pedina si ferma nel momento in cui ha toccato tutti i nodi almeno una volta. Sia X la variabile aleatoria che indica il nodo su cui si ferma (ossia il nodo che viene visitato per ultimo). Osservate che X può essere uno qualunque dei nodi $u \in \{1, 2, \ldots, n\}$ (non può essere il nodo 0, perchè questo risulta già visitato all'istante $t = 0$). Lo scopo di questo esercizio è scoprire qual è la distribuzione di probabilità di X. Prima di procedere provate a intuire la soluzione: riflettete su quali potrebbero essere i nodi con una probabilità grande che la pedina di fermi su di loro, quali quelli con una probabilità piccola.

1. Scrivere un programma, in un linguaggio di programmazione a piacere, che simuli il processo. Il programma deve prendere in input un numero intero positivo n, deve simulare il processo usando un generatore di numeri casuali, e deve restituire in output il nodo $u \in \{1, \ldots, n\}$ su cui si ferma la pedina.
2. Usando il programma del punto precedente, scrivere un programma che prenda in input due numeri, n e t, simuli il processo su un anello di $n + 1$ nodi per t volte, e per ogni nodo $u \in \{1, \ldots, n\}$ restituisca il numero di volte che la pedina si è fermata sul nodo u. Il programma deve leggere l'input da un file `input.txt` che contiene una sola riga con due numeri separati da uno spazio, il primo numero è n, il secondo numero è t. Il programma deve scrivere un file `output.txt` con n righe, sulla riga u-esima deve esserci il numero di volte che la pedina si è fermata sul nodo u.
3. Eseguite il programma del punto precedente con t abbastanza grande e per vari valori di n (per esempio, $t = 10^5$ e $n = 5, 10, 15, 20$). Alla luce dei risultati che ottenete, formulate una congettura su quale dovrebbe essere la distribuzione di probabilità della variabile X.
4. Provate a dimostrare la vostra congettura.