"ALGORITMI"

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2014/15

1^a prova in itinere – 13 gennaio 2015

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1

Si risolva l'equazione di ricorrenza

$$T(n) = \frac{27}{a} \cdot T\left(\frac{n}{a}\right) + \Theta\left(n^2 \log n\right)$$

al variare del parametro reale $a > 1$.

ESERCIZIO 2

Si ordinino le funzioni $n^2 \log n$, $\log^2 n$, 2^n, $4^{\log n}$ per tasso di crescita.

ESERCIZIO 3

(a) Si stabilisca se l'array $[25, 12, 13, 6, 9, 7, 5, 1, 8, 4]$ è un max-heap.

(b) Si descriva la procedura MAX-HEAPIFY e quindi si illustri l'azione di MAX-HEAPIFY$(A, 3)$ sull'array $A = [1, 3, 9, 2, 2, 14, 12, 1, 1, 1, 1, 10, 11, 11, 9]$.

ESERCIZIO 4

Si descriva l'algoritmo COUNTING SORT (campo di applicazione, pseudocodice, complessità, proprietà, ecc.) e lo si illustri sull'array $A = [2, 0, 3, 6, 2, 0, 4, 2]$.

ESERCIZIO 5

Si illustri un semplice algoritmo che risolva in tempo lineare il problema della selezione per un'arbitraria statistica d'ordine, basato su una subroutine data, MEDIAN, che trova la mediana in tempo lineare.

ESERCIZIO 6

(a) Sia T una tabella hash di dimensione 16, inizialmente vuota, organizzata con il metodo dell'indirizzamento aperto. Sia $h(x, i) : \mathbb{N} \times \{0, 1, \ldots, 15\} \to \{0, 1, \ldots, 15\}$ la funzione hash quadratica definita da

$$h(x, i) = \left(x + \frac{i(i+1)}{2}\right) \mod 16\,.$$

Si illustri l'inserimento delle chiavi 84, 6, 116, 18, 100, 97, 96, 113, 22, 7, 10, 71 (nell'ordine dato) nella tabella T utilizzando la funzione hash h.

(b) Si enunci l'ipotesi di *hashing uniforme* e si forniscano dei limiti superiori al numero medio di scansioni in ricerche *con* e *senza* successo in una tabella hash con fattore di carico α, assumendo l'ipotesi di hashing uniforme.

(c) La funzione $h(x, i)$ definita sopra soddisfa l'ipotesi di hashing uniforme? Perchè?

"ALGORITMI"

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2015/16

1^a prova in itinere – 17 dicembre 2015

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

(A) Si enuncino il Teorema Master e il suo Corollario, quindi si risolva la seguente equazione di ricorrenza al variare del parametro $b \geq 1$:

$$T(n) = 16 \cdot T\left(\frac{n}{2b}\right) + \Theta\left(n^2 \log^2 n\right) .$$

Per quali valori di b si ha: (a) $T(n) = \mathcal{O}(n^4)$; (b) $T(n) = \Omega(n^4)$; (c) $T(n) = \Theta(n^4)$; (d) $T(n) = \mathcal{O}(n^2 \log n)$?

(B) Si ordinino per tasso di crescita le funzioni $n^2 \log n$, $n \log^2 n$, $\frac{2^n}{n^4}$, $n^2 \log^n n$, $n \log^4 n$.

ESERCIZIO 2 (Foglio A)

(a) Si enunci l'ipotesi di *hashing uniforme* e si forniscano dei limiti superiori al numero medio di scansioni in ricerche *con* e *senza* successo in una tabella hash con fattore di carico α, assumendo l'ipotesi di hashing uniforme.

(b) (Facoltativo) Si descriva la procedura per l'inserimento di una chiave in una tabella hash organizzata con l'indirizzamento aperto.

(c) Data la funzione $h(x, i) =_{Def} (x + 3i) \mod 17$, si illustri l'inserimento delle chiavi

23, 43, 48, 52, 21, 5, 78, 55, 35, 62, 72, 17, 51, 58, 46

in una tabella hash di dimensione 17, inizialmente vuota e organizzata con l'indirizzamento aperto, utilizzando $h(x, i)$ come funzione hash.

ESERCIZIO 3 (Foglio B)

(a) Si fornisca (con dimostrazione) un limite superiore sull'altezza di un heap binario con n elementi.

(b) Si descriva la procedura Build-Max-Heap e se ne illustri l'azione sull'array $A = [5, 4, 1, 3, 20, 12, 14, 15, 10, 8]$.

ESERCIZIO 4 (Foglio B)

Si descriva l'algoritmo Counting Sort (campo di applicazione, pseudocodice, complessità, proprietà, ecc.) e lo si illustri sull'array di coppie $A = [(9, A), (6, B), (0, C), (9, D), (5, E), (7, F), (0, G), (9, H), (5, I)]$, da ordinare rispetto alla prima componente.

"ALGORITMI"

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2014/15

Terza sessione di esami (II appello) - 06 ottobre 2015

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Equazione di ricorrenza)

Si enuncino il Teorema Master ed il suo Corollorio, quindi si risolva la seguente equazione di ricorrenza al variare del parametro $\alpha \geq 1$:

$$T(n) = \alpha \cdot T\left(\frac{n}{2}\right) + n^2 \log^2 n\,.$$

Per quali valori di α si ha: (a) $T(n) = \mathcal{O}(n^3)$; (b) $T(n) = \Omega(n^2 \log^3 n)$; (c) $T(n) = \Omega(n^2 \log^4 n)$?

ESERCIZIO 2 (Alberi "rosso-neri")

(a) Si illustri la struttura dati degli alberi rosso-neri.

(b) Si definisca l'*altezza nera* di un nodo in un albero rosso-nero. Quindi si enunci una minorazione del numero di nodi interni in un sottoalbero radicato in un nodo x di un albero rosso-nero e la si utilizzi per dimostrare un limite superiore all'altezza di un albero rosso-nero con n nodi interni.

ESERCIZIO 3 (Visita in profondità)

Sia dato il grafo orientato $\mathcal{G}$ con insieme di vertici $V = \{A, B, C, D, E, F, G\}$ e i cui archi sono rappresentati dalle seguenti liste di adiacenza:

A → B, C, D	C → A	F → D, E
B → C, D	D → E, F	

Dopo aver descritto l'algoritmo di visita in profondità, si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A, indicando i tempi di inizio e fine visita per ciascun vertice, e la classificazione di tutti gli archi (es. archi d'albero, all'indietro, ecc.). Si rappresenti inoltre la foresta DFS ottenuta.

(Facoltativo) Si definisca la nozione di *componente fortemente connessa* (cfc) di un grafo orientato e si descriva un algoritmo per calcolare le cfc di un grafo orientato. Quindi si determinino le cfc del grafo $\mathcal{G}$.

ESERCIZIO 4 (Tavole hash)

(a) Data la funzione $h(x, i) =_{Def} (x + 4i) \mod 19$, si illustri l'inserimento delle chiavi

23, 43, 21, 5, 62, 72, 58, 48, 52, 46, 78, 55, 35, 17, 51

in una tabella hash di dimensione 17, inizialmente vuota e organizzata con l'indirizzamento aperto, utilizzando $h(x, i)$ come funzione hash.

(b) Si enunci l'ipotesi di *hashing uniforme*, si forniscano dei limiti superiori al numero medio di scansioni in ricerche *con* e *senza* successo in una tabella hash con fattore di carico α, assumendo l'ipotesi di hashing uniforme.

ESERCIZIO 5 (Programmazione dinamica)

Si enunci in dettaglio il problema della moltiplicazione di una sequenza di matrici. Quindi, utilizzando la metodologia della programmazione dinamica, si illustri una soluzione della variante del problema della moltiplicazione di una sequenza di matrici in cui si è interessati a massimizzare il numero di prodotti scalari, piuttosto che a minimizzarlo.

Qual è la complessità dell'algoritmo trovato in funzione della lunghezza della sequenza di matrici ?

“ALGORITMI”

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2014/15

Prolungamento terza sessione di esami - 17 dicembre 2015

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

(A) Si enuncino il Teorema Master e il suo Corollario, quindi si risolva la seguente equazione di ricorrenza al variare del parametro $b \geq 1$:

$$T(n) = 16 \cdot T\left(\frac{n}{2b}\right) + \Theta\left(n^2 \log^2 n\right) .$$

Per quali valori di b si ha: (a) $T(n) = \mathcal{O}(n^4)$; (b) $T(n) = \Omega(n^4)$; (c) $T(n) = \Theta(n^4)$; (d) $T(n) = \mathcal{O}(n^2 \log n)$?

(B) Si ordinino per tasso di crescita le funzioni $n^2 \log n$, $n \log^2 n$, $\frac{2^n}{n^4}$, $n^2 \log^n n$, $n \log^4 n$.

ESERCIZIO 2 (Foglio A)

(a) Si enunci l'ipotesi di *hashing uniforme* e si forniscano dei limiti superiori al numero medio di scansioni in ricerche *con* e *senza* successo in una tabella hash con fattore di carico α, assumendo l'ipotesi di hashing uniforme.

(b) Si descriva la procedura per l'inserimento di una chiave in una tabella hash organizzata con l'indirizzamento aperto.

ESERCIZIO 3 (Foglio A)

Sia T un testo di 1000 caratteri in un alfabeto con dieci caratteri $c_1, \ldots, c_{10}$, le cui frequenze sono rispettivamente

10, 10, 30, 30, 30, 70, 100, 120, 200, 400.

Dopo aver definito la nozione di *codice prefisso*, si determini il numero minimo di bit necessari per rappresentare il testo T utilizzando un codice prefisso ottimo, illustrando anche l'algoritmo utilizzato.

ESERCIZIO 4 (Foglio B)

Si descriva l'algoritmo Counting Sort (campo di applicazione, pseudocodice, complessità, proprietà, ecc.) e lo si illustri sull'array di coppie $A = [(9, A), (6, B), (0, C), (9, D), (5, E), (7, F), (0, G), (9, H), (5, I)]$, da ordinare rispetto alla prima componente.

ESERCIZIO 5 (Foglio B)

Sia dato il grafo orientato $\mathcal{G}$ con insieme di vertici $V = \{A, B, C, D, E, F, G, H, I, L\}$ e i cui archi sono rappresentati dalle seguenti liste di adiacenza:

A → B, L	C → B, D	E → F	G → F	I → G, L
B → C	D → F, I	F → H	H → E, G	L → D

(a) Dopo aver descritto l'algoritmo di visita in profondità, si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A, indicando i tempi di inizio e fine visita per ciascun vertice, e la classificazione di tutti gli archi (es. archi d'albero, all'indietro, ecc.). Si rappresenti inoltre la foresta DFS ottenuta.

(b) Si definisca la nozione di *componente fortemente connessa* (cfc) di un grafo orientato e si descriva un algoritmo per calcolare le cfc di un grafo orientato. Quindi si determinino le cfc del grafo $\mathcal{G}$.

“ALGORITMI”

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2015/16

Prima sessione di esami (I appello) – 08 febbraio 2016

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

Si descriva la struttura dati del *max-heap binario* e si fornisca, con dimostrazione, un limite superiore stretto per l'altezza di un heap binario con n nodi.

Quindi si descrivano le procedure Max-Heapify e Build-Max-Heap (anche mediante il loro pseudo-codice) e si illustri la procedura Build-Max-Heap sull'array $C := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$.

Infine si descriva l'algoritmo Heapsort.

ESERCIZIO 2 (Foglio A)

Si consideri il grafo orientato $\mathcal{G}$ rappresentato dalle seguenti liste di adiacenza:

A → E, G	D → A	G → B, D
B → C	E → F	H → C, E
C → G	F → B, D	

(a) Dopo aver descritto l'algoritmo di visita in profondità (anche con pseudo-codice), si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A (e poi procedendo lessicograficamente), indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi.

(b) Si definisca la nozione di *componenti fortemente connesse* (cfc) di un grafo orientato e si descriva un algoritmo per il loro calcolo, indicandone la complessità computazionale. Quindi si determinino le componenti fortemente connesse del grafo $\mathcal{G}$, utilizzando i risultati della visita già effettuata per il punto precedente.

ESERCIZIO 3 (Foglio B)

Sia T un testo di 500 caratteri in un alfabeto con i simboli $a_1, \ldots, a_6$, le cui frequenze sono rispettivamente

30, 50, 60, 60, 100, 120, 200.

Dopo aver definito la nozione di *codice prefisso*, si determini il numero minimo di bit necessari per rappresentare il testo T utilizzando un codice prefisso ottimo, illustrando l'algoritmo utilizzato (anche mediante pseudo-codice). Qual è il risparmio percentuale rispetto ad una codifica minimale di T con un codice a lunghezza fissa?

ESERCIZIO 4 (Foglio B)

(A) Si enuncino il Teorema Master e il suo Corollario, quindi si risolva la seguente equazione di ricorrenza al variare del parametro reale $a \geq 1$:

$$T(n) = a \cdot T\left(\frac{n}{2}\right) + \Theta\left(n^3 \log^3 n\right) .$$

Per quali valori di a si ha: (i) $T(n) = \mathcal{O}(n^3 \log^3 n)$; (ii) $T(n) = \Omega(n^3 \log^5 n)$; (iii) $T(n) = \Theta(n^4)$?

(B) Si ordinino per tasso di crescita le funzioni n, $\frac{n}{\log^2 n}$, $\frac{\log^3 n}{n}$, $\log n$, $\frac{n^2}{\log^4 n}$.

“ALGORITMI”

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2015/16

Prima sessione di esami (II appello) – 29 febbraio 2016

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

Dopo aver enunciato il problema della selezione, si descriva (fornendone anche lo pseudo-codice) un algoritmo efficiente per la sua soluzione, valutandone anche la complessità computazionale.

ESERCIZIO 2 (Foglio A)

Si consideri il grafo orientato $\mathcal{G}$ rappresentato dalle seguenti liste di adiacenza:

A → B, H	D → E, G	G → D, H
B → C, D	E → F, G	H → D, F
C → A, F	F → D, G	

(A) Si descriva l'algoritmo di visita in profondità, fornendone anche lo pseudo-codice e determinandone la complessità computazionale. Quindi si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A (e poi procedendo lessicograficamente), indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi.

(B) Si definiscano le *componenti fortemente connesse* di un grafo orientato e si descriva un algoritmo per il loro calcolo.

(C) Si determinino le componenti fortemente connesse del grafo $\mathcal{G}$.

ESERCIZIO 3 (Foglio B)

(A) Si enuncino il Teorema Master e il suo Corollario.

(B) Si definiscano le notazioni asintotiche $\mathcal{O}(f(n))$, $\Omega(f(n))$, $\Theta(f(n))$, $\omega(f(n))$ per una data funzione $f : \mathbb{N} \to \mathbb{N}$.

(C) Si risolva l'equazione di ricorrenza $T(n) = 4 \cdot T\left(\frac{n}{2}\right) + \Theta\left(n^c \log^2 n\right)$ al variare del parametro reale $c \geq 1$.

(D) Sia $T(n)$ la funzione di cui al punto precedente. Per quali valori di c si ha:

(i) $T(n) = \Theta(n)$; (ii) $T(n) = \mathcal{O}(n^2)$; (iii) $T(n) = \Omega(n^2)$; (iv) $T(n) = \omega(n^2)$?

ESERCIZIO 4 (Foglio B)

Si enunci in dettaglio il problema della moltiplicazione di una sequenza di matrici. Quindi, utilizzando la metodologia della programmazione dinamica, si illustri una soluzione della variante del problema della moltiplicazione di una sequenza di matrici in cui si è interessati a massimizzare il numero di prodotti scalari, piuttosto che a minimizzarlo. Qual è la complessità dell'algoritmo trovato in funzione della lunghezza della sequenza di matrici ?

“ALGORITMI”

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2015/16

Seconda sessione di esami (I appello) – 20 giugno 2016

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

Si descriva l'algoritmo Counting Sort (campo di applicazione, pseudocodice, complessità, proprietà, ecc.) e lo si illustri sull'array di coppie $A = [(7, A), (8, B), (0, C), (9, D), (8, E), (7, F), (0, G), (9, H), (8, I), (0, L)]$, da ordinare rispetto alla prima componente.

ESERCIZIO 2 (Foglio A)

Si consideri il grafo orientato $\mathcal{G}$ rappresentato dalle seguenti liste di adiacenza:

A → B, C	D → C	G → D
B → A	E → D	H → G
C → D, E, F	F → G, H	

(A) Si descriva l'algoritmo di visita in profondità, fornendone anche lo pseudo-codice e determinandone la complessità computazionale. Quindi si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A (e poi procedendo lessicograficamente), indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi.

(B) Si definiscano le *componenti fortemente connesse* di un grafo orientato e si descriva un algoritmo per il loro calcolo.

(C) Si determinino le componenti fortemente connesse del grafo $\mathcal{G}$.

ESERCIZIO 3 (Foglio B)

(A) Si enuncino il Teorema Master e il suo Corollario.

(B) Si definiscano le notazioni asintotiche $\mathcal{O}(f(n))$, $\Omega(f(n))$, $\Theta(f(n))$, $o(f(n))$ per una data funzione $f : \mathbb{N} \to \mathbb{N}$.

(C) Si risolva l'equazione di ricorrenza $T(n) = a \cdot T\left(\frac{n}{3}\right) + \Theta\left(n^2 \log^2 n\right)$ al variare del parametro reale $a \geqslant 1$.

(D) Sia $T(n)$ la funzione di cui al punto precedente. Per quali valori del parametro a si ha $n^2 \log^2 n = o(T(n))$?

ESERCIZIO 4 (Foglio B)

Si consideri la seguente operazione $\oplus$ sui numeri naturali, definita da: $a \oplus b =_{Def} 3a + 5b$.

(a) Si verifichi con un esempio a scelta che l'operazione$\oplus$ non è associativa.

(b) Utilizzando la metodologia della programmazione dinamica, si determini un algoritmo che, data una sequenza di numeri naturali $a_1, a_2, \ldots, a_n$, calcoli il valore *massimo* che l'espressione $a_1 \oplus a_2 \oplus \cdots \oplus a_n$ può assumere al variare di tutte le possibili parentesizzazioni.

“ALGORITMI”

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2015/16

Seconda sessione di esami (II appello) – 06 luglio 2016

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

(a) Si illustri la struttura dati del max-heap binario mettendola anche in relazione con la sua rappresentazione con array.

(b) Si descrivano le procedure MAX-HEAPIFY e BUILD-MAX-HEAP e si illustri l'azione di BUILD-MAX-HEAP sulla sequenza di interi $[2, 4, 10, 2, 2, 11, 13, 3, 15, 3]$.

(c) Si descriva l'algoritmo Heapsort.

ESERCIZIO 2 (Foglio A)

Si consideri il grafo orientato $\mathcal{G}$ rappresentato dalle seguenti liste di adiacenza:

A → B	D → A	G → C, D, F
B → D	E → F, G	H → D, F
C → A, E, H	F → B, H	

(A) Si descriva l'algoritmo di visita in profondità, fornendone anche lo pseudo-codice e determinandone la complessità computazionale. Quindi si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A (e poi procedendo lessicograficamente), indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi.

(B) Si definiscano le *componenti fortemente connesse* di un grafo orientato e si descriva un algoritmo per il loro calcolo.

(C) Si determinino le componenti fortemente connesse del grafo $\mathcal{G}$.

ESERCIZIO 3 (Foglio B)

(A) Si enuncino il Teorema Master e il suo Corollario.

(B) Si definiscano le notazioni asintotiche $\mathcal{O}(f(n))$, $o(f(n))$, $\Theta(f(n))$ per una data funzione $f\colon \mathbb{N} \to \mathbb{N}$.

(C) Si risolva l'equazione di ricorrenza $T(n) = 64 \cdot T\left(\frac{n}{b}\right) + \Theta\left(n^3 \log^3 n\right)$ al variare del parametro reale $b > 1$.

(D) Sia $T(n)$ la funzione di cui al punto precedente. Si stabilisca per quali valori del parametro b si ha

(i) $T(n) = \mathcal{O}(n^3 \log^3 n)$; (ii) $T(n) = o(n^3 \log^3 n)$; (iii) $T(n) = \Theta(n^6)$.

ESERCIZIO 4 (Foglio B)

Nel contesto della metodologia *greedy*, si enunci il problema di ottimizzazione relativo alla *selezione di attività* e se ne discuta una soluzione efficiente, valutandone la complessità computazionale e illustrandola sul seguente insieme $S = \{a_1, \ldots, a_{10}\}$ di attività, caratterizzate dai seguenti tempi iniziali e finali:

i	1	2	3	4	5	6	7	8	9	10
s_i	2	4	13	12	5	1	10	11	7	6
f_i	5	7	14	13	10	6	12	12	9	9

"ALGORITMI"

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA
ANNO ACCADEMICO 2015/16

Terza sessione di esami (I appello) – 19 settembre 2016

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

Si descriva l'algoritmo Counting Sort (campo di applicazione, pseudocodice, complessità, proprietà, ecc.) e lo si illustri sull'array di coppie $A = [(5, A), (8, B), (2, C), (9, D), (8, E), (5, F), (2, G), (9, H), (8, I), (2, L)]$, da ordinare rispetto alla prima componente.

ESERCIZIO 2 (Foglio A)

Si consideri il grafo orientato $\mathcal{G}$ rappresentato dalle seguenti liste di adiacenza:

A → E, F, G	D → G	G → C, H
B → C, E	E → B, G	H → C, D
C → D	F → A, B, H	

(A) Si descriva l'algoritmo di visita in profondità, fornendone anche lo pseudo-codice e determinandone la complessità computazionale. Quindi si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A (e poi procedendo lessicograficamente), indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi.

(B) Si definiscano le *componenti fortemente connesse* di un grafo orientato e si descriva un algoritmo per il loro calcolo. Quindi si determinino le componenti fortemente connesse del grafo $\mathcal{G}$.

ESERCIZIO 3 (Foglio B)

(A) Si enuncino il Teorema Master e il suo Corollario.

(B) Si definiscano le notazioni asintotiche $\mathcal{O}(f(n))$, $\Theta(f(n))$ e $\Omega(f(n))$ per una data funzione $f : \mathbb{N} \to \mathbb{N}$.

(C) Si risolva l'equazione di ricorrenza $T(n) = 9 \cdot T\left(\frac{n}{b}\right) + \Theta\left(n^2 \log^3 n\right)$ al variare del parametro reale $b > 1$.

(D) Sia $T(n)$ la funzione di cui al punto precedente. Si stabilisca per quali valori del parametro b si ha
(i) $T(n) = \mathcal{O}(n^3)$; (ii) $T(n) = \Theta(n^2 \log^2 n)$; (iii) $T(n) = \Omega(n^2 \log^4 n)$.

ESERCIZIO 4 (Foglio B)

Sia T un testo di 270 caratteri in un alfabeto con i sette caratteri A, B, C, D, E, F, G, le cui frequenze sono rispettivamente

40, 20, 10, 10, 60, 60, 70.

Dopo aver definito la nozione di *codice prefisso*, si determini il numero minimo di bit necessari per rappresentare il testo T utilizzando un codice prefisso ottimo, illustrando anche dettagliatamente l'algoritmo utilizzato.
Qual il fattore di compressione ottenuto, relativamente ad una codifica a 3 bit a lunghezza fissa?

“ALGORITMI”

CORSO DI STUDIO IN INFORMATICA (laurea triennale)

UNIVERSITÀ DEGLI STUDI DI CATANIA

ANNO ACCADEMICO 2015/16

Terza sessione di esami (II appello) – 05 ottobre 2016

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Foglio A)

Si descriva la struttura dati del *max-heap binario* e si fornisca, con dimostrazione, un limite superiore stretto per l'altezza di un heap binario con n nodi.

Quindi si descrivano le procedure MAX-HEAPIFY e BUILD-MAX-HEAP (anche mediante il loro pseudo-codice) e si illustri la procedura BUILD-MAX-HEAP sull'array $C := [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]$.

Infine si descriva l'algoritmo HEAPSORT.

ESERCIZIO 2 (Foglio A)

Si consideri il grafo orientato $\mathcal{G}$ rappresentato dalle seguenti liste di adiacenza:

A → E, G	D → C, F	G → B, C, D
B → G	E → H	H → A
C → F	F → D	

(A) Si descriva l'algoritmo di visita in profondità, fornendone anche lo pseudo-codice e determinandone la complessità computazionale. Quindi si effettui la visita in profondità del grafo $\mathcal{G}$ a partire dal vertice A (e poi procedendo lessicograficamente), indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi.

(B) Si definiscano le *componenti fortemente connesse* di un grafo orientato e si descriva un algoritmo per il loro calcolo. Quindi si determinino le componenti fortemente connesse del grafo $\mathcal{G}$.

ESERCIZIO 3 (Foglio B)

Si consideri la seguente operazione $\oplus$ sui numeri naturali, definita da: $a \oplus b =_{Def} 2a + 3b$.

(a) Si verifichi con un esempio a scelta che l'operazione $\oplus$ non è associativa.

(b) Utilizzando la metodologia della programmazione dinamica, si determini un algoritmo che, data una sequenza di numeri naturali $a_1, a_2, \ldots, a_n$, calcoli il valore *massimo* che l'espressione $a_1 \oplus a_2 \oplus \cdots \oplus a_n$ possa assumere al variare di tutte le possibili parentesizzazioni.

ESERCIZIO 4 (Foglio B)

(a) Data la funzione $h(x, i) =_{Def} (x + 2i) \mod 17$, si illustri l'inserimento delle chiavi

$$55,\ 38,\ 59,\ 87,\ 68,\ 70,\ 62,\ 52,\ 33,\ 47,\ 79,\ 30$$

in una tabella hash di dimensione 17, inizialmente vuota e organizzata con l'indirizzamento aperto, utilizzando $h(x, i)$ come funzione hash.
Quante collisioni si sono verificate?

(b) Si enunci l'ipotesi di *hashing uniforme*. Quindi si forniscano dei limiti superiori al numero medio di scansioni in ricerche *con* e *senza* successo in una tabella hash con fattore di carico α, assumendo l'ipotesi di hashing uniforme.