Algoritmi e Strutture Dati **Anno Accademico 2018–2019**

Primo Appello
Seconda Prova del 12 Giugno 2019

ATTENZIONE: L'uso di **break** e **continue** è fortemente **deprecato** e comporta una penalità di **1 punto ad ogni utilizzo.**

Esercizio 1. (5 punti)
In un grafo diretto, un nodo viene detto *sorgente* se non ha archi entranti (ovvero ha grado entrante uguale a zero). Dato un grafo diretto $G = (V, E)$ rappresentato mediante *matrice di adiacenza*, scrivere lo pseudo-codice di un algoritmo che restituisca il numero di sorgenti presenti nel grafo G.

Esercizio 2. (5 + 6 + 6 punti)
Sia dato un albero binario T in cui ciascun nodo è bianco oppure nero. Un *sottoalbero monocolore* è caratterizzato dall'avere tutti i nodi dello stesso colore (o bianchi o neri). Indicando le scelte implementative necessarie per poter determinare il colore di un singolo nodo, si fornisca una soluzione di costo computazionale lineare nel numero dei nodi di T ai seguenti problemi:

1. (5 punti) Dato T in input, restituire il numero totale di nodi bianchi presenti nell'albero.
2. (6 + 6 punti) **IN ALTERNATIVA con l'esercizio 3.2** Dato T in input, restituire la dimensione massima di un sottoalbero monocolore di T. L'algoritmo deve essere prima **descritto**[1] esaurientemente[2] a parole (6 punti) e successivamente deve essere fornito lo pseudo-codice (6 punti).

Le soluzioni devono prima essere spiegate brevemente a parole e successivamente deve essere fornito lo pseudo-codice.

Esercizio 3. (8 + 6 + 6 punti)
Dato un array A di n interi *distinti* ed *ordinati*, un *elfo domestico* dispettoso decide di scambiare una coppia di elementi distinti in A. Risolvere i due problemi seguenti:

1. (8 punti) Dato A, fornire lo pseudo-codice di algoritmo che, in tempo lineare $O(n)$, individui gli indici i e j di A in cui è avvenuto lo scambio.
2. (6 + 6 punti) **IN ALTERNATIVA con l'esercizio 2.2** Dato A e la posizione i di uno degli elementi scambiati, (6 punti) **descrivere**[1] esaurientemente[2] a parole un algoritmo che individui la posizione dell'altro indice j coinvolto nello scambio in tempo $O(\log n)$. (6 punti) Successivamente, fornire lo pseudo-codice.

[1] La descrizione può essere anche per punti.
[2] Per *esauriente* si intende che, oltre all'idea, devono essere specificati anche i dettagli significativi (es. puntatori nulli, estremi dell'array, casi di valori particolari che potrebbero creare problemi, etc...); specificare, se ne vengono usate, come sono fatte strutture ausiliarie; etc..

Algoritmi e Strutture Dati **Anno Accademico 2018–2019**

Secondo Appello
Seconda Prova del 26 Giugno 2019

ATTENZIONE: L'uso di **break** e **continue** è fortemente **deprecato** e comporta una penalità di **1 punto ad ogni utilizzo.**

Esercizio 1. (8 + 3 + 1 punti)
Dato un DAG (Grafo Diretto Aciclico) G rappresentato con una matrice di adiacenza, risolvere i seguenti esercizi:

1. (8 punti) Descrivere un algoritmo che restituisce `TRUE` se il DAG è un albero binario completo[3] (non necessariamente bilanciato) in cui gli archi sono diretti dai padri ai figli, e `FALSE` altrimenti.

   L'algoritmo deve essere brevemente spiegato a parole e, successivamente, deve essere fornito lo pseudocodice[4].

2. (3 punti) Dimostrare la correttezza dell'algoritmo descritto nel punto precedente.

3. (1 punto) Spiegare **a parole** le modifiche da apportare all'algoritmo affinché questo restituisca 0 se il DAG non è una foresta di alberi binari completi, oppure il numero k di alberi binari completi nella foresta (e si avrà $k \geq 1$).

Esercizio 2. (8 + 1 punti)
Dato un grafo diretto $G = (V, E)$ e tre vertici distinti $u, v, w \in V$, risolvere i seguenti esercizi:

1. (8 punti) Descrivere un algoritmo che restituisce `TRUE` se esiste un cammino in G che parte da u e arriva a v passando per w, e `FALSE` altrimenti.

   L'algoritmo deve essere brevemente spiegato a parole e, successivamente, deve essere fornito lo pseudocodice[5].

2. (1 punto) Studiare il costo computazionale dell'algoritmo proposto.

Esercizio 3 - IN ALTERNATIVA ALL'ESERCIZIO 4. (5 punti - punteggio massimo raggiungibile 26)
Dato un albero binario T implementato con nodi e puntatori, scrivere un algoritmo di costo computazionale lineare nel numero dei nodi che scambia il figlio destro con il figlio sinistro di ogni nodo dell'albero.

Esercizio 4 - IN ALTERNATIVA ALL'ESERCIZIO 3. (8 + 1 punti - punteggio massimo raggiungibile 30)
(8 punti) Sia T un albero binario qualsiasi, implementato con nodi e puntatori, in cui ogni nodo ha, tra gli altri, un campo *key* che contiene un intero positivo. Dato T e un valore intero positivo k, si descriva un algoritmo che restituisce `TRUE` se esiste in T un cammino radice-foglia tale per cui la somma dei valori contenuti nei nodi del cammino sia pari a k, e `FALSE` altrimenti. (1 punto) Studiare il costo computazionale dell'algoritmo.

Descrivere brevemente a parole l'algoritmo proposto e, successivamente, fornire lo pseudo-codice.

[3] In un albero binario completo ogni nodo ha zero o due figli.
[4] Nello pseudo-codice deve essere utilizzata la matrice di adiacenza.
[5] Lo psudo-codice può essere scritto scegliendo una rappresentazione del grafo tra liste o matrice di adiacenza, **oppure** ad alto livello (utilizzando espressioni del tipo $v \in V$ o $(u, v) \in E$). L'importante è essere coerenti con la scelta fatta.

Algoritmi e Strutture Dati **Anno Accademico 2018–2019**

Terzo Appello
Seconda Prova del 18 Luglio 2019

ATTENZIONE: L'uso di **break** e **continue** è fortemente **deprecato** e comporta una penalità di **1 punto ad ogni utilizzo.**

Esercizio 1 - IN ALTERNATIVA ALL'ESERCIZIO 2. (7 punti)
Dato un array A di n interi, descrivere mediate DIAGRAMMA DI FLUSSO un algoritmo che restituisce `TRUE` se la sequenza è ordinata (in ordine non decrescente o non crescente), `FALSE` altrimenti.

Esercizio 2 - IN ALTERNATIVA ALL'ESERCIZIO 1. (6 punti (+ 2 punti))
Dato un array A di $n \geq 2$ interi (non necessariamente positivi), fornire lo pseudocodice di un algoritmo che restituisce il valore del massimo prodotto ottenibile moltiplicando due valori distinti nell'array. (+ 2 punti) se il costo computazionale dell'algoritmo è $O(f(n))$ con $f(n) < n^2$.

Esercizio 3. (2 + 2 + 8 + 1 punti)

Dato un min-heap H di $n \geq 2$ elementi, un *folletto* modifica il valore della chiave memorizzata in due nodi distinti (non noti): una chiave viene sostituita con un valore strettamente maggiore di quello originario, l'altra viene sostituita con un valore strettamente minore di quello originario. Rispondere ai seguenti quesiti:

1. (2 punti) Fornire un esempio, con $n \geq 10$, in cui, dopo l'intervento del folletto, H non è più un heap;
2. (2 punti) Fornire un esempio, con $n \geq 10$, in cui, dopo l'intervento del folletto, H è ancora un heap;
3. (8 punti) Supponendo che l'heap sia implementato con un array, fornire lo pseudo-codice di un algoritmo che restituisce `TRUE` se dopo l'intervento del folletto H è ancora un heap, e `FALSE` altrimenti. Per semplicità si assuma che l'array che memorizza l'heap abbia dimensione n;
4. (1 punto) Studiare il costo computazionale dell'algoritmo proposto al punto precedente.

Esercizio 3. (2 + 3 + 8 + 1 punti) Sia A un array di n numeri naturali tale che esiste $m \geq 1$ intero per cui $n = 2^m - 1$. Si consideri il problema di costruire un albero binario di ricerca perfettamente bilanciato le cui chiavi siano i valori nell'array A. Rispondere ai seguenti quesiti:

1. (2 punti) Fornire un esempio, scegliendo un $m \geq 3$, di input del problema e corrispondete output;
2. (3 punti) Descrivere a parole un algoritmo che risolve il problema, motivando le scelte fatte;
3. (8 punti) Fornire lo pseudo-codice dell'algoritmo proposto al punto precedente;
4. (1 punto) Studiare il costo computazionale dell'algoritmo proposto.

Algoritmi e Strutture Dati **Anno Accademico 2018–2019**

Quarto Appello
Seconda Prova del 17 Settembre 2019

ATTENZIONE: L'uso di **break** e **continue** è fortemente **deprecato** e comporta una penalità di **1 punto ad ogni utilizzo.**

Esercizio 1. (2 + 7 + 3 punti)
Sia dato un array A contenente n interi *distinti e ordinati* (in ordine crescente). Si consideri il problema di determinare se esiste un indice i per cui $A[i] = i$.

(a) **(2 punti)** Fornire un esempio di istanza in cui tale indice i esiste, e uno in cui tale indice non esiste.

(b) **(7 punti)** Fornire lo pseudocodice di un algoritmo di costo computazionale $O(\log n)$ che restituisca un indice i tale che $A[i] = i$. Se tale indice non esiste, l'algoritmo restituisce -1.

(c) **(3 punti)** Motivare le scelte fatte per risolvere il problema.

Esercizio 2. (2 + 8 + 1 + 1 punti)
Dato un grafo $G = (V, E)$, si definisce *triangolo* un insieme di tre nodi *distinti* di V tali che esista in E un arco tra ogni coppia di nodi nell'insieme, ovvero, un triangolo è un insieme di nodi $\{x, y, z\} \in V$ tali che $(x, y), (y, z), (z, x) \in E$. Si assuma che gli identificativi dei nodi del grafo siano gli interi positivi $1, \ldots, n$, con $n = |V|$.

(a) **(2 punti)** Dato un grafo G con 10 nodi, non orientato e completo, elencare tutti i triangoli in G che sono composti esclusivamente da nodi con identificativo dispari.

(b) **(8 punti)** Descrivere a parole e fornire lo pseudocodice di un algoritmo che, preso in input un grafo non orientato G (qualunque) implementato mediante matrice di adiacenza, restituisce il numero dei triangoli presenti in G composti esclusivamente da nodi con identificativo dispari.

(c) **(1 punto)** Studiare il costo computazionale dell'algoritmo.

(d) **(1 punto)** Quanti sono, in generale, i triangoli composti esclusivamente da nodi con identificativo dispari in un grafo completo non diretto con n nodi? Qual è il minimo n affichè esista almeno un tale trinagolo?

Esercizio 3. (1 + 6 + 1 punti)
L'*altezza* di un albero è il numero di archi sul cammino radice-foglia più lungo presente nell'albero. Un albero *ternario* è un albero in cui ogni nodo può avere da zero a tre figli.

(a) **(1 punto)** Descrivere come rappresentare, mediante nodi e puntatori, un albero ternario.

(b) (**6 punti**) Utilizzando la descrizione precedente, scrivere lo pseudocodice di un algoritmo *ricorsivo* che prende in input un albero ternario T e restituisce l'altezza di T. Scrivere a parole quale sia il caso base e come procede l'algoritmo nel caso base e nel caso non base.

(c) (**1 punto**) Studiare il costo computazionale dell'algoritmo proposto.

Quinto Appello
Seconda prova del 22 Gennaio 2020

ATTENZIONE: L'uso di **break** e **continue** è fortemente **deprecato** e comporta una penalità di **1 punto ad ogni utilizzo.**

Esercizio 1. (6 + 1 + 2 punti)

(a) **(+6 punti)** Dato un albero binario T, scrivere lo pseudocodice di un algoritmo ricorsivo che restituisce il numero di nodi dell'albero che hanno esattamente un figlio. Descrivere esplicitamente a parole il caso base e il caso non base, indicando come procedere in entrambi i casi.

(b) **(+1 punto)** Studiare il costo computazionale dell'algoritmo proposto.

(c) **(+2 punti)** Fare un esempio di albero binario di altezza almeno tre ed elencare, nell'ordine in cui avvengono, le chiamate ricorsive effettuate dall'algoritmo proposto al punto (a), eseguito sull'albero preso ad esempio. Indicare, a fianco di ogni chiamata ricorsiva, il risultato della chiamata stessa.

Esercizio 2. (1 + 1 + 6/10 punti)
Dato un grafo *orientato* (diretto) $G = (V, E)$, un nodo $v \in V$ si dice di tipo *Roma* se da ogni altro vertice $w \in V$ si può raggiungere il nodo v con un cammino orientato che parte da w e arriva a v.

(a) **(+ 1 punto)** Mostrare un esempio di grafo con un nodo tipo *Roma*.

(b) **(+ 1 punto)** Mostrare un esempio di grafo dove non esiste un nodo tipo *Roma*.

(c) **(+ 6/10 punti)** Descrivere a parole e fornire lo pseudocodice di un algoritmo che, preso in input un grafo orientato $G = (V, E)$ e un nodo $v \in V$, restituisce **true** se v è di tipo *Roma* e **false** altrimenti. Si studi il costo computazionale dell'algoritmo proposto. Il punteggio assegnato alla risposta sarà di massimo **10 punti** se il costo computazionale risulta essere (correttamente) $O(|V| + |E|)$, oppure di massimo **6 punti** se il costo risulta essere maggiore.

Esercizio 3. (1 + 6 + 2 punti)
Sia dato un array A contenente n interi composto dalla concatenazione di due sequenze monotone (una delle due possibilmente vuota), la prima non-decrescente e la seconda non-crescente.

(a) **(+ 1 punto)** Mostrare **due** esempi: un array (con almeno 8 elementi) in cui le due sequenze monotone non sono vuote, e uno (con almeno 8 elementi) in cui una delle due sequenze monotone è vuota.

(b) **(+ 6 punti)** Scrivere lo pseudocodice di un algoritmo di complessità lineare che ordina l'array A.

(c) **(+2 punti)** Spiegare perché l'algoritmo proposto è corretto.

Sesto Appello
Seconda prova del 14 Febbraio 2020

ATTENZIONE: L'uso di **break** e **continue** è fortemente **deprecato** e comporta una penalità di **1 punto ad ogni utilizzo.**

Esercizio 1. (1 + 8 + 2 punti) Sia dato un array A contenente una sequenza di n interi *distinti* che è il risultato di uno shift circolare di k posizioni a destra di una sequenza ordinata in ordine crescente, per $k \in [0, n-1]$.

1. **(1 punto)** Scelto un $n \geq 10$, mostrare un esempio di un'istanza di A con $k = 0$ e un'altra con $k = \lceil n/3 \rceil$.
2. **(8 punti)** Descrivere a parole e fornire lo pseudocodice di un algoritmo, di costo computazionale in tempo sublineare, che dato A determini e restituisca il valore minimo in A.
3. **(2 punti)** Rispondere alla seguente domanda, argomentando la risposta: l'algoritmo funziona anche se i valori in A *non* sono distinti?

Esercizio 2. (8 + 1 + 1 + 2 punti) Sia M una matrice quadrata $n \times n$ tale che $n = 2^k$, per $k \geq 0$, e $M[i, j] \in \{0, 1\}$, per ogni $i, j = 0, \ldots, n-1$.

1. **(8 punti)** Descrivere e fornire lo pseudcodice di un algoritmo basato sulla tecnica DIVIDE-et-IMPERA che, dati M e n, restituisce il numero di uni nella matrice.
2. **(1 punto)** Studiare il costo computazionale dell'algoritmo proposto.
3. **(1 punto)** Elencare le prime 6 chiamate effettuale dalla funzione ricorsiva dell'algoritmo proposto al punto 1 su input una generica matrice quadrata 16×16.
4. **(2 punti)** Spiegare come modificare l'algoritmo affinché risolvera il problema anche nel caso in cui la matrice in input sia rettangolare $n \times m$, per $n, m \geq 1$.

Esercizio 3. (2 + 7 + 1 punti) Dato un grafo $G = (V, E)$, con $V = \{1, \ldots, n\}$ ed un array $A[1..n]$ contenente n interi, G è detto **ben colorato** se per ogni $(i, j) \in E$ vale che $A[i] = A[j]$.

1. **(2 punti)** Si fornisca un esempio di grafo ben colorato e un esempio di grafo non ben colorato.
2. **(7 punti)** Si descriva a parole e si fornisca lo pseudocidce di un'algoritmo per il seguente problema:

   Input: grafo $G = (V, E)$ *diretto*, rappresentato mediante *liste di adiacenza*, e un array $A[1..n]$ contenente n interi;

   Output: `TRUE` se G è *ben colorato*, `FALSO` altrimenti
3. **(1 punto)** Si studi il costo computazionale dell'algoritmo.