Domanda #1, quiz-loop

☐ **1) [1 punteggio, quiz-loop-1]** Si tira quanto vale la variabile **i** quando **x0** vale **sqrt(0)** e **x1** vale **sqrt(1)**

☐ **2) [1 punteggio, quiz-loop-2]** Quale tra **x0** e **x1** vale **sqrt(2)**? Si tira quanto vale **i** quando **x0** vale **sqrt(2)** e **x1** vale **sqrt(3)**

☐ **3) [1 punteggio, quiz-loop-3]** Si tira quanto vale **i** quando **x0** vale **sqrt(n)** e **x1** vale **sqrt(n+1)**

n **x0** **x1** **i**

1) LA CONDIZIONE NEL CICLO MIRA A VERIFICARE CHE $x_1 \geq n$ MA È UN CICLO FINITO IL PUNTO D'ARRIVO IL MASSIMO DEL PUNTO CHE POSSO PRENDERE SENZA FARE SUPERARE IL $\sqrt{n}$.
VISTO CHE IL PUNTO DI ARRIVO MASSIMO DEVE VERIFICARE $n^2 \geq n$ SICCOME CHE $n \geq 1$ IL PUNTO DI ARRIVO È LINEARE $O(n)$.
QUINDI CHIAMO CHE IL MIGLIORE APPROCCIO SIA QUELLO DI ESEGUIRE LOOP IL PUNTO DI ARRIVO È ARRIVO DI OPERAZIONE INFATTI SE OGNI SINGOLO OPERAZIONE DURA MASSIMO AL LINEARE $\Rightarrow$ SI EFFETTUA 2^i CHIAMATE A PUNTO CHIAMO IN UNA RICERCA DI BINARY SEARCH $O(\log n)$.
SE CON LINEARE PARTO: $2^i = n \Rightarrow$ DIVENTA CHE DIVENTA $\Theta(n)$.

2)
87 77 33 29 23 49 59 76

PARTENDO DI NUOVO NEL CASO PEGGIORE SCANSIONO OGNI SINGOLO ELEMENTO DI NUOVO O IL MASSIMO DEVO SCANSIONARE LA PRIMA VOLTA CHIAMATA.

a) INSERIAMO IL MINORE (DUNQUE 23):
(77) 87 33 29 23 49 59 76
*PRIMO PIVOT

b) INSERIAMO IL MAGGIORE (87):
(77) 33 29 23 49 59 (87)

c) INSERIAMO IL MINORE (23):
(77) (23) 33 29 49 59 (87)

d) INSERIAMO IL MINORE (33):
(77) (23) (29) 33 49 59 (87)

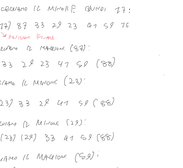
e) INSERIAMO IL MAGGIORE (59):
(77) (23) (29) 33 49 (59) (87)

f) INSERIAMO IL MINORE (49)
(77) (23) (29) (33) 49 (59) (87)

g) INSERIAMO NUOVA 2010 49 CHE È IL PUNTO FINALE
SEGNAMO PIVOT: (77) (23) (29) (33) (49) (59) (87)

Domanda #2, binary search tree

B



A

[Domanda 2.1] Si descriva in modo generale - cioè senza fare riferimento all'esempio riportato a fianco - l'algoritmo detettivo su Binary Search Tree. Se l'algoritmo non è stato ancora trovato analizzate, si spieghi anche il loro funzionamento.

[Domanda 2.2 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D2.1 è corretto)] Si scriva il codice di tale algoritmo. Si usi il linguaggio preferito, purché sia comprensibile da chiunque.

[Domanda 2.3] Si spieghi dettagliatamente, mediante testo e disegni chiari, i passaggi della rotazione destra e sinistra. Si spieghi anche le eventuali rotazioni a doppiette mediante, ovviamente sempre testo e disegni.

C

2) LA PRIMA DOMANDA DEBBAMO SI OCCUPA DI RICHIAMARE UN NODO ATTRAVERSO DALL'ALBERO/MANCHIAMO LE PROPRIETÀ DEL BST.
ABBIAMO 3 CASI PARTICOLARI:

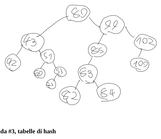
1) IL NODO DA ELIMINARE È UNA FOGLIA (NESSUN FIGLIO)
ELIMINO SEMPLICEMENTE IL NODO

2) IL NODO DA ELIMINARE HA UN SOLO FIGLIO:
SI ELIMINA IL NODO E SI COLLEGA IL FIGLIO DEL NODO ELIMINATO AL GENITORE DEL NODO ELIMINATO

3) IL NODO DA ELIMINARE HA DUE FIGLI
INTRODUCO DEVO SOSTITUIRE IL NODO PRESCELTO NEL NODO DA CANCELLARE CON IL VALORE MINIMO DEL SOTTOALBERO DESTRO
UTILIZZO LA FUNZIONE ANALOGA DELETE MIN (DELETE MAX VALIDAMENTE CASO) CHE SELEZIONA LA COPPIA CHIAVE_VALORE ED ELIMINA IL NODO
QUESTA COPPIA CHIAVE_VALORE VIENE PASSATA ALLA DELETE BIN ED ESEGUO LO SWAP

b) LA DELETE HA COMPLESSITÀ $\Theta(n)$ DOVE n SANREBBE LUNGHEZZA DELL'ALBERO.
SAREBBE IL CASO PEGGIORE IN QUANTO DOBBIAMO ATTRAVERSARE LA FOGLIA ALBERO.

c) AVEVAMO A DISPOSIZIONE IL VALORE MINIMO NEL SOTTO ALBERO DESTRO (IN QUESTO CASO IL NODO CON VALORE MINORE È 80.
LA FUNZIONE DELETE MIN SALVA LA COPPIA CHIAVE_VALORE DEL NODO 80 E PASSA I VALORI ALLA DELETE BIN ED ELIMINA IL NODO 80. DOPO LA DELETE BIN SOSTituisce IL 55 CON 80



Domanda #3, tabelle di hash

Si consideri una tabella di hash con linee di collisione che implementa un dizionario **D** in cui le chiavi sono sequenze di 4 cifre decimali che indichiamo con **a, b, c, d** e i valori sono stringhe.
La tabella di hash ha 7 bucket indicizzati da 0 a 6 e la funzione di hash è **h(a,b,c,d) = (a+b+c+d) mod 7**.

[Preparazione dei dati, senza sono MA LA PRESENZA DI DUE O PIU' ERRORI NEL CALCOLO DELLA FUNZIONE DI HASH COMPORTA L'ASSEGNAZIONE DI ZERO PUNTI ALL'INTERO ESERCIZIO]

Completate su questo foglio lo schema mostrato sotto calcolando la funzione di hash delle chiavi scritte nella prima riga e riportando nella corrispondente colla nella terza riga.

chiave = 3741	chiave = 1147	chiave = 2271	chiave = 7252	chiave = 2163	chiave = 7208	chiave = 1311	chiave = 2789
valore = "ona"	valore = "cor"	valore = "gra"	valore = "pe"	valore = "le"	valore = "ne"	valore = "cor"	valore = "gio"
h(chiave) = ∅	h(chiave) = 6	h(chiave) = ∅	h(chiave) = 2	h(chiave) = ∅	h(chiave) = 3	h(chiave) = 6	h(chiave) = ∅

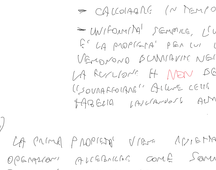
[Domanda 3.1, 1/3 del punteggio] Il dizionario D implementato dalla tabella di hash è inizialmente vuoto.

☐ **D3.1:** Disegnate la tabella di hash che si ottiene dagli inserimenti delle stringhe "ona", "cor", "gra", "pe", "le", "ne", "cor", "gio" associate alle chiavi su quattro cifre riportate nella schema sopra, nell'ordine in cui compaiono nello schema da sinistra verso destra. Nel caso ci fossero delle **chiavi** duplicate, ignorate l'inserimento e scrivete esplicitamente sul foglio "operata coppia chiave-valore è stata ignorata perché la chiave è duplicata".

☐ **[Domanda 3.2, 1/3 del punteggio (la risposta sarà valutata solo se la risposta alla domanda 3.1 è corretta)]**
☐ **D3.2.1:** Quali sono le due principali "buone proprietà" che una funzione di hash deve avere?
☐ **D3.2.2:** La funzione di hash **h** applicata alle chiavi nello schema disegnato sopra gode della prima di queste due "buone proprietà"? Motivare esaurientemente la risposta.
☐ **D3.2.3:** La funzione di hash **h** applicata alle chiavi nello schema disegnato sopra gode della seconda di queste due "buone proprietà"? Motivare esaurientemente la risposta.

NON SERVE PER FARE QUESTO ESERCIZIO È PARTE DI CONTROLLARE IL NUMERO DEL MODULO (IN QUESTO CASO 7)
SE "CONTANO" NEI CALCOLI
QUINDI 3741 SUCCEDE DOPO $3^2 + 4 + 1 = 6$
DOVREMMO IL NUMERO NEL H

1)



NEW AGGIUNGO CHIAVI DUPLICATE

2) a) LE DUE QUALITÀ PROPRIETÀ SONO:
- CALCOLABILE IN TEMPO COSTANTE
- UNIFORMITÀ SEMPRE, L'UNIFORMITÀ È LA PROPRIETÀ PER CUI LE CHIAVI VERRANNO DISTRIBUITE NEI BUCKET. LA FUNZIONE H NON DEVE "DOVREGGIONARE" ALCUNE CELLE DELLA TABELLA LASCIANDO ALTRE VUOTE

b) LA PRIMA PROPRIETÀ VIENE ASSICURATA IN QUANTO LE OPERAZIONI ARITMETICHE COME SOMMA, MOLTIPLICAZIONE E MODULO SONO CON COMPLESSITÀ OPERAZIONI CALCOLABILI IN TEMPO COSTANTE $(\Theta(1))$
INOLTRE LE CHIAVI SONO COMPOSTE DOPO SA DI CIFRE CHE VANNO DA 0 A 9 QUINDI QUALSIASI OPERAZIONE DI COMPOSTO CI IMPIEGA CIRCA LO STESSO TEMPO.

c) NO, L'UNIFORMITÀ NON È ASSICURATA IN QUANTO ALCUNE CELLE SONO VUOTE MENTRE ALTRE HANNO UN NUMERO DI CHIAVI DIVERSO

[Domanda 3.3, 1/3 del punteggio] Sia **m** la dimensione della tabella e sia **n** il numero di elementi nella tabella. Si assuma che **m** ed **n** siano entrambi diversi da 0.
☐ **D3.3.1:** Si descriva l'algoritmo per l'inserimento di un elemento data la sua chiave; l'algoritmo deve esplicitamente e correttamente gestire il caso di chiavi duplicate.
☐ **D3.3.2 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto):** Sotto quali condizioni l'inserimento ha complessità Theta(n)?
☐ **D3.3.3 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto):** Sotto quali condizioni l'inserimento ha complessità Theta(m/n)?
☐ **D3.3.4 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto):** Sotto quali condizioni l'inserimento ha complessità Theta(1)?

1) L'INSERIMENTO SI DIVIDE IN 3 PASSAGGI:

1) CALCOLO DELLE INDICI:
VARIANDO IL MI PERMETTE DI TRASFORMARE LA CHIAVE DELL'ELEMENTO IN UN NUMERO, QUESTO NUMERO MI RAPPRESENTA DIVIDUALE DOVE L'ELEMENTO DOVRÀ ESSERE INSERITO NELLA TABELLA

b) VERIFICA DELLA POSIZIONE
SE LA POSIZIONE CALCOLATA È VUOTA ALLORA L'ELEMENTO VIENE INSERITO DIRETTAMENTE SI
SE POSIZIONE GIÀ OCCUPATA ALLORA SI VERIFICA UNA COLLISIONE

c) GESTIONE CHIAVI DUPLICATE
POSSO ALLORA UN CONTROLLO ACCURATO DELLA MIA FUNZIONE CHE IN CASO DI CHIAVI DUPLICATE MI RESTITUISCA UN ERRORE

2) INSERIMENTO VALE $\Theta(m)$ QUANDO GLI ELEMENTI SI TROVANO NELLO STESSO INDICE

3) UNIFORMITÀ SEMPRE

4) NEL CASO IN CUI L'ELEMENTO È GIÀ PRESENTE IL QUINDI L'INSERIMENTO "FALLISCE" OPPURE DETTAGLIATAMENTE NEL CASO DI TABELLA VUOTA O INSERIMENTO IN TESTA