Tecnologie e architetture per la gestione dei dati

Prova parziale — 14 aprile 2022 — Compito A

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ **Nome** _____ **Matricola** _____

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 1 (15%)
Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione *R(A,C,....)* di cardinalità pari circa a $L = 10.000.000$, con ennuple di $l = 40$ byte e campo A chiave di tipo stringa (ad esempio, il codice fiscale) e campo C di tipo intero anch'esso chiave (ad esempio, un numero di matricola) — quindi la relazione ha due chiavi, ognuna delle quali, da sola, identifica univocamente le ennuple. Supporre che il sistema offra

- strutture primarie di tre tipi: (a) disordinate, (b) ordinate rispetto ad un campo su cui è definito un indice, (c) hash
- indici di tipo B-tree (anche più di uno sulla stessa relazione, primari o secondari)

Considerare un carico applicativo con le seguenti operazioni **tutte di lettura**

1. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) della chiave A, con frequenza giornaliera $f_1 = 100.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $s = 10$ ennuple;
2. ricerca di una ennupla sulla base del valore (completo) della chiave C, con frequenza giornaliera $f_2 = 100.000$;
3. scansione dell'intera relazione, ordinata sulla base del valore di A, con frequenza giornaliera $f_3 = 10$;
4. scansione dell'intera relazione, ordinata sulla base del valore di C, con frequenza giornaliera $f_4 = 1$;

Progettare l'organizzazione fisica della relazione, (i) scegliendo la struttura primaria fra le varie possibilità e (ii) individuando gli eventuali indici. Ragionare in termini di numero di accessi a memoria secondaria, assumendo che (1) gli indici primari abbiano profondità $p_1 = 3$ e gli indici secondari profondità $p_2 = 4$, (2) il buffer disponibile abbia (2a) dimensione minore del numero di blocchi del file e maggiore della radice quadrata di tale numero e (2b) permetta di mantenere stabilmente in memoria due livelli di indice, sia per i primari sia per i secondari; (3) l'hash, per gli accessi puntuali, abbia costo unitario. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descrizione struttura			
Costo operazione 1			
Costo operazione 2			
Costo operazione 3			
Costo operazione 4			
Costo totale			

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 2 (15%)
Considerare ancora il caso illustrato nella domanda precedente, ma con riferimento ad una realtà con le seguenti frequenze per le operazioni:

1. $f_1 = 10.000$
2. $f_2 = 1.000.000$
3. $f_3 = 1$
4. $f_3 = 1$

Considerare ancora almeno due alternative (quelle che intuitivamente si ritengono migliori in questo caso).

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descrizione struttura			
Costo operazione 1			
Costo operazione 2			
Costo operazione 3			
Costo operazione 4			
Costo totale			

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 3 (30%)
Come noto, in molti sistemi (ad esempio in PostgreSQL) è possibile ordinare fisicamente le relazioni, senza che però l'ordinamento venga mantenuto. Allo scopo, l'ordinamento viene rieseguito periodicamente, su dati che sono in parte ordinati e in parte no. Una possibile strategia seguita (molto comune) è la seguente:

- le eliminazioni vengono realizzate semplicemente "marcando" i record e lasciando quindi lo spazio inutilizzato
- gli inserimenti vengono effettuati in coda, nell'ordine in cui vengono richiesto
- le modifiche vengono trattate ciascuna come un'eliminazione e un inserimento

Di conseguenza, si possono presentare situazioni come quella mostrata in sotto, in cui la relazione è in parte ordinata e in parte disordinata (l'asterisco "*" indica i record cancellati). Si noti che la relazione presenta molti blocchi ordinati (dodici nell'esempio) e alcuni disordinati (tre).

Supponendo di avere a disposizione **sei** pagine buffer, descrivere brevemente nel riquadro l'algoritmo da utilizzare per riordinare la relazione, mostrando anche il contenuto dei buffer in occasione del primo caricamento e quello alla fine dell'ordinamento. Rispondere anche alle altre domande che vengono poste.

112	...
120	...
142	...
182	...
200	...
222	...
232	...
251	...
345	...
*	
443	...
501	...
502	...
525	...
686	...
*	
735	...
774	...
783	...
801	...
805	...
839	...
842	...
900	...

535	...
171	...
484	...
838	...
262	...
646	...

- Descrivere sinteticamente l'algoritmo (bastano poche righe informali, non serve pseudocodice)

- Con riferimento all'esempio e all'algoritmo proposto,
  quanti blocchi vengono letti? [ ] E quanti ne vengono scritti? [ ]

- Mostrare il contenuto delle pagine di buffer al primo caricamento e il contenuto delle stesse alla fine dell'esecuzione dell'algoritmo

Primo caricamento delle pagine del buffer	Contenuto finale delle pagine del buffer

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 4 (20%)
Considerare un sistema con dischi con le seguenti caratteristiche

- tempo medio di posizionamento della testina (tempo di seek) $t_S = 4$ msec
- tempo medio di latenza (attesa dovuta alla rotazione) $t_L = 3$ msec
- tempo minimo di lettura di un blocco $t_B = 10$ μsec

Rispondere alle seguenti domande mostrando formula e valore numerico

1. Qual è il tempo medio necessario per leggere un blocco del quale sia dato l'indirizzo fisico?

2. Qual è il tempo medio necessario per la scansione sequenziale di un file costituito da $F = 200$ blocchi contigui, non letti di recente?

3. Qual è il tempo che si può ipotizzare necessario per eseguire quattro accessi diretti a record di un file attraverso un indice che abbia profondità $p = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 50$, non usato di recente?

4. Qual è il tempo che si può ipotizzare necessario per eseguire $m = 100.000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p = 4$, fan-out $f_I = 50$, con disponibilità di circa $P = 4000$ pagine di buffer?

Domanda 5 (20%)
Si consideri un B-tree con nodi intermedi che contengono due chiavi e tre puntatori e foglie con due chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 50, 59, 7, 32, 11, 27, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Tecnologie e architetture per la gestione dei dati

Prova parziale — 14 aprile 2022 — Compito B

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ **Nome** _____ **Matricola** _____

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito B

Domanda 1 (15%)
Si considerino un sistema con blocchi di dimensione $B = 8000$ byte e una relazione *R(A,C,....)* di cardinalità pari circa a $N = 10.000.000$, con ennuple di $l = 80$ byte e campo A chiave di tipo stringa (ad esempio, il codice fiscale) e campo C di tipo intero anch'esso chiave (ad esempio, un numero di matricola) — quindi la relazione ha due chiavi, ognuna delle quali, da sola, identifica univocamente le ennuple. Supporre che il sistema offra

- strutture primarie di tre tipi: (a) disordinate, (b) ordinate rispetto ad un campo su cui è definito un indice, (c) hash
- indici di tipo B-tree (anche più di uno sulla stessa relazione, primari o secondari)

Considerare un carico applicativo con le seguenti operazioni **tutte di lettura**

1. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) della chiave A, con frequenza giornaliera $f_1 = 100.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $n = 10$ ennuple;
2. ricerca di una ennupla sulla base del valore (completo) della chiave C, con frequenza giornaliera $f_2 = 100.000$;
3. scansione dell'intera relazione, ordinata sulla base del valore di A, con frequenza giornaliera $f_3 = 10$;
4. scansione dell'intera relazione, ordinata sulla base del valore di C, con frequenza giornaliera $f_4 = 1$;

Progettare l'organizzazione fisica della relazione, (i) scegliendo la struttura primaria fra le varie possibilità e (ii) individuando gli eventuali indici. Ragionare in termini di numero di accessi a memoria secondaria, assumendo che (1) gli indici primari abbiano profondità $p_1 = 3$ e gli indici secondari profondità $p_2 = 4$, (2) il buffer disponibile abbia (2a) dimensione minore del numero di blocchi del file e maggiore della radice quadrata di tale numero e (2b) permetta di mantenere stabilmente in memoria due livelli di indice, sia per i primari sia per i secondari; (3) l'hash, per gli accessi puntuali, abbia costo unitario. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descrizione struttura			
Costo operazione 1			
Costo operazione 2			
Costo operazione 3			
Costo operazione 4			
Costo totale			

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito B

Domanda 2 (15%)
Considerare ancora il caso illustrato nella domanda precedente, ma con riferimento ad una realtà con le seguenti frequenze per le operazioni:

1. $f_1 = 10.000$
2. $f_2 = 1.000.000$
3. $f_3 = 1$
4. $f_3 = 1$

Considerare ancora almeno due alternative (quelle che intuitivamente si ritengono migliori in questo caso).

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descrizione struttura			
Costo operazione 1			
Costo operazione 2			
Costo operazione 3			
Costo operazione 4			
Costo totale			

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito B

Domanda 3 (30%)
Come noto, in molti sistemi (ad esempio in PostgreSQL) è possibile ordinare fisicamente le relazioni, senza che però l'ordinamento venga mantenuto. Allo scopo, l'ordinamento viene rieseguito periodicamente, su dati che sono in parte ordinati e in parte no. Una possibile strategia seguita (molto comune) è la seguente:

- le eliminazioni vengono realizzate semplicemente "marcando" i record e lasciando quindi lo spazio inutilizzato
- gli inserimenti vengono effettuati in coda, nell'ordine in cui vengono richiesto
- le modifiche vengono trattate ciascuna come un'eliminazione e un inserimento

Di conseguenza, si possono presentare situazioni come quella mostrata in sotto, in cui la relazione è in parte ordinata e in parte disordinata (l'asterisco "*" indica i record cancellati). Si noti che la relazione presenta molti blocchi ordinati (dodici nell'esempio) e alcuni disordinati (tre).

Supponendo di avere a disposizione **sei** pagine buffer, descrivere brevemente nel riquadro l'algoritmo da utilizzare per riordinare la relazione, mostrando anche il contenuto dei buffer in occasione del primo caricamento e quello alla fine dell'ordinamento. Rispondere anche alle altre domande che vengono poste.

111	...
120	...
141	...
181	...
200	...
221	...
232	...
251	...
345	...
*	
443	...
501	...
502	...
525	...
686	...
*	
735	...
774	...
783	...
801	...
805	...
839	...
842	...
900	...

535	...
171	...
484	...
838	...
262	...
646	...

- Descrivere sinteticamente l'algoritmo (bastano poche righe informali, non serve pseudocodice)

- Con riferimento all'esempio e all'algoritmo proposto,
  quanti blocchi vengono letti? [ ] E quanti ne vengono scritti? [ ]

- Mostrare il contenuto delle pagine di buffer al primo caricamento e il contenuto delle stesse alla fine dell'esecuzione dell'algoritmo

Primo caricamento delle pagine del buffer	Contenuto finale delle pagine del buffer

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito B

Domanda 4 (20%)
Considerare un sistema con dischi con le seguenti caratteristiche

- tempo medio di posizionamento della testina (tempo di seek) $t_S = 4$ msec
- tempo medio di latenza (attesa dovuta alla rotazione) $t_L = 4$ msec
- tempo minimo di lettura di un blocco $t_B = 10$ μsec

Rispondere alle seguenti domande mostrando formula e valore numerico

1. Qual è il tempo medio necessario per leggere un blocco del quale sia dato l'indirizzo fisico?

2. Qual è il tempo medio necessario per la scansione sequenziale di un file costituito da $F = 200$ blocchi contigui, non letti di recente?

3. Qual è il tempo che si può ipotizzare necessario per eseguire quattro accessi diretti a record di un file attraverso un indice che abbia profondità $p = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 50$, non usato di recente?

4. Qual è il tempo che si può ipotizzare necessario per eseguire $m = 200.000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p = 4$, fan-out $f_I = 50$, con disponibilità di circa $P = 4000$ pagine di buffer?

Domanda 5 (20%)
Si consideri un B-tree con nodi intermedi che contengono due chiavi e tre puntatori e foglie con due chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 38, 51, 8, 32, 21, 25, 70, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Tecnologie e architetture per la gestione dei dati

Prova parziale — 14 aprile 2022 — Compito A

Cenni sulle soluzioni
(per il compito A, gli altri sono simili, le varianti del testo sono in rosso)

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ **Nome** _____ **Matricola** _____

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 1 (15%)
Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione *R(A,C,....)* di cardinalità pari circa a $L = 10.000.000$, con ennuple di $l = 40$ byte e campo A chiave di tipo stringa (ad esempio, il codice fiscale) e campo C di tipo intero anch'esso chiave (ad esempio, un numero di matricola) — quindi la relazione ha due chiavi, ognuna delle quali, da sola, identifica univocamente le ennuple. Supporre che il sistema offra

- strutture primarie di tre tipi: (a) disordinate, (b) ordinate rispetto ad un campo su cui è definito un indice, (c) hash
- indici di tipo B-tree (anche più di uno sulla stessa relazione, primari o secondari)

Considerare un carico applicativo con le seguenti operazioni **tutte di lettura**

1. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) della chiave A, con frequenza giornaliera $f_1 = 100.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $s = 10$ ennuple;
2. ricerca di una ennupla sulla base del valore (completo) della chiave C, con frequenza giornaliera $f_2 = 100.000$;
3. scansione dell'intera relazione, ordinata sulla base del valore di A, con frequenza giornaliera $f_3 = 10$;
4. scansione dell'intera relazione, ordinata sulla base del valore di C, con frequenza giornaliera $f_4 = 1$;

Progettare l'organizzazione fisica della relazione, (i) scegliendo la struttura primaria fra le varie possibilità e (ii) individuando gli eventuali indici. Ragionare in termini di numero di accessi a memoria secondaria, assumendo che (1) gli indici primari abbiano profondità $p_1 = 3$ e gli indici secondari profondità $p_2 = 4$, (2) il buffer disponibile abbia (2a) dimensione minore del numero di blocchi del file e maggiore della radice quadrata di tale numero e (2b) permetta di mantenere stabilmente in memoria due livelli di indice, sia per i primari sia per i secondari; (3) l'hash, per gli accessi puntuali, abbia costo unitario. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. strutt.	Struttura ordinata su A con indice primario e indice secondario su C	Struttura ordinata su C con indice primario e indice secondario su A	Struttura hash su C e indice secondario su A
Costo Op. 1	p_1-2+1=ca.2: visita indice (p_1-2) e 1 blocco con i 10 record	p_2-2+10=ca.12: visita indice (p_2-2) e 10 record	p_2-2+10=ca.12: visita indice (p_2-2) e 10 record
Costo Op. 2	p_2-2+1=ca.3: visita indice (p_2-2) e 1 record	p_1-2+1=ca.2: visita indice (p_1-2) e 1 record	1
Costo Op. 3	Essendo il file ordinato, basta la scansione: L/(B/c)=100.000	È necessario ordinare il file, servono due passate: 3 × L/(B/c)=300.000	Serve ordinamento: 300.000
Costo Op. 3	Serve ordinamento: 300.000	Basta scansione: 100.000	Serve ordinamento: 300.000
Tot	2 × 100.000 + 3 × 100.000 + 100.000 × 10 + 300.000 × 1 = ca 1.800.000	12 × 100.000 + 2 × 100.000 + 300.000 × 10 + 100.000 × 1 = ca 4.500.000	12 × 100.000 + 1 × 100.000 + 300.000 × 10 + 300.000 × 1 = ca 4.600.000

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 2 (15%)
Considerare ancora il caso illustrato nella domanda precedente, ma con riferimento ad una realtà con le seguenti frequenze per le operazioni:

1. $f_1 = 10.000$
2. $f_2 = 1.000.000$
3. $f_3 = 1$
4. $f_3 = 1$

Considerare ancora almeno due alternative (quelle che intuitivamente si ritengono migliori in questo caso).

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. strutt.	Struttura ordinata su A con indice primario e indice secondario su C	Struttura ordinata su C con indice primario e indice secondario su A	Struttura hash su C e indice secondario su A
Costo Op. 1	2	12	12
Costo Op. 2	3	2	1
Costo Op. 3	100.000	300.000	300.000
Costo Op. 3	300.000	100.000	300.000
Tot	$2 \times 10.000 + 3 \times 1.000.000 + 100.000 \times 1 + 300.000 \times 1$ = ca 3.400.000	$12 \times 10.000 + 2 \times 1.000.000 + 300.000 \times 10 + 100.000 \times 1$ = ca 2.500.000	$12 \times 10.000 + 1 \times 1.000.000 + 300.000 \times 10 + 300.000 \times 1$ = ca 1.700.000

Domanda 3 (30%)
Come noto, in molti sistemi (ad esempio in PostgreSQL) è possibile ordinare fisicamente le relazioni, senza che però l'ordinamento venga mantenuto. Allo scopo, l'ordinamento viene rieseguito periodicamente, su dati che sono in parte ordinati e in parte no. Una possibile strategia seguita (molto comune) è la seguente:

- le eliminazioni vengono realizzate semplicemente "marcando" i record e lasciando quindi lo spazio inutilizzato
- gli inserimenti vengono effettuati in coda, nell'ordine in cui vengono richiesto
- le modifiche vengono trattate ciascuna come un'eliminazione e un inserimento

Di conseguenza, si possono presentare situazioni come quella mostrata in sotto, in cui la relazione è in parte ordinata e in parte disordinata (l'asterisco "*" indica i record cancellati). Si noti che la relazione presenta molti blocchi ordinati (dodici nell'esempio) e alcuni disordinati (tre).

Supponendo di avere a disposizione **sei** pagine buffer, descrivere brevemente nel riquadro l'algoritmo da utilizzare per riordinare la relazione, mostrando anche il contenuto dei buffer in occasione del primo caricamento e quello alla fine dell'ordinamento. Rispondere anche alle altre domande che vengono poste.

112	...
120	...
142	...
182	...
200	...
222	...
232	...
251	...
345	...
*	
443	...
501	...
502	...
525	...
686	...
*	
735	...
774	...
783	...
801	...
805	...
839	...
842	...
900	...

535	...
171	...
484	...
838	...
262	...
646	...

- Descrivere sinteticamente l'algoritmo (bastano poche righe informali, non serve pseudocodice)
  Risposta: Si può utilizzare una variante del merge sort che tenga conto del fatto che una frazione significativa del file è ordinata. Si può procedere nel modo seguente:
  1. caricare in tre pagine del buffer i tre blocchi disordinati
  2. ordinare i record in queste tre pagine
  3. eseguire un merge fra queste tre pagine e i (dodici) blocchi ordinati, che vengono caricati uno alla volta in una quarta pagina del buffer; una quinta pagina viene utilizzata per accumulare via via il risultato, con scrittura quando essa è piena; durante la fusione vengono anche trascurati i record cancellati, e così il numero di blocchi scritti è 14

  In questo modo, ciascuno dei blocchi originari viene letto una sola volta e ciascuno dei blocchi del file riordinato viene scritto una sola volta.

- Con riferimento all'esempio e all'algoritmo proposto,
  quanti blocchi vengono letti? 15 E quanti ne vengono scritti? 14

- Mostrare il contenuto delle pagine di buffer al primo caricamento e il contenuto delle stesse alla fine dell'esecuzione dell'algoritmo

Primo caricamento delle pagine del buffer

535	...
171	...
484	...
838	...
262	...
646	...

Contenuto finale delle pagine del buffer

171	...
262	...
484	...
535	...
646	...
838	...
842	...
900	...
842	...
900	...

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 4 (20%)
Considerare un sistema con dischi con le seguenti caratteristiche

- tempo medio di posizionamento della testina (tempo di seek) $t_S = 4$ msec
- tempo medio di latenza (attesa dovuta alla rotazione) $t_L = 3$ msec
- tempo minimo di lettura di un blocco $t_B = 10$ μsec

Rispondere alle seguenti domande mostrando formula e valore numerico

1. Qual è il tempo medio necessario per leggere un blocco del quale sia dato l'indirizzo fisico?

La somma del tempo medio di seek, del tempo medio di latenza e del tempo minimo di lettura di un blocco

$$t_{tot} = t_S + t_L + t_B = 4 + 3 + 0,01 \text{ msec } = \text{ ca } 7 \text{ msec}$$

2. Qual è il tempo medio necessario per la scansione sequenziale di un file costituito da $F = 200$ blocchi contigui, non letti di recente?

Il tempo medio necessario per leggere un blocco (il primo) più il tempo minimo di lettura per ciascuno degli altri

$$t_{tot} + (F - 1) \times t_B = \text{ ca } 9 \text{ msec}$$

3. Qual è il tempo che si può ipotizzare necessario per eseguire quattro accessi diretti a record di un file attraverso un indice che abbia profondità $p = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 50$, non usato di recente?

Si può immaginare all'inizio nessuno nodo dell'indice si trovi nel buffer. Quindi, per il primo record, $p = 4$ accessi per l'indice e uno per il record del file, per gli altri la radice si trova nel buffer, quindi $p - 1$ più uno:

$$((p + 1) + 3 \times ((p - 1) + 1)) \times t_{tot} = \text{ ca } 120 \text{ msec}$$

4. Qual è il tempo che si può ipotizzare necessario per eseguire $m = 100.000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p = 4$, fan-out $f_I = 50$, con disponibilità di circa $P = 4000$ pagine di buffer?

Si può immaginare che la radice e altri due livelli dell'indice restino nel buffer, dopo il primo caricamento, quindi, per ogni record, un accesso all'indice e uno al file, in posizioni non prevedibili (qualche accesso in più per il caricamento iniziale e qualcuno in meno per blocchi acceduti più volte):

$$(p - 3 + 1) \times m \times t_{tot} = \text{ ca } 1400 \text{ sec}$$

Tecnologie e architetture per la gestione dei dati — 14 aprile 2022 — Compito A

Domanda 5 (20%)
Si consideri un B-tree con nodi intermedi che contengono due chiavi e tre puntatori e foglie con due chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 50, 59, 7, 32, 11, 27, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Vengono mostrate le soluzioni per il compito A. Le altre sono analoghe .