

Basi di dati II

Prova parziale — 8 aprile 2019 — Compito A

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ Nome _____ Matricola _____

Basi di dati II — 8 aprile 2019 — Compito A

Domanda 1 (20%)

Considerare la relazione sotto schematizzata, definita su vari attributi, uno dei quali è la chiave, i cui valori sono mostrati. Supporre che la relazione abbia un fattore di blocco pari a 2 (e quindi occupi 15 blocchi) e che siano disponibili 8 buffer. Considerare l'esecuzione di un mergesort a più vie (e due passate) sulla relazione e mostrare lo stato delle strutture in memoria centrale e secondaria dopo l'esecuzione di cinque chiamate al metodo `next()` sullo scan che implementa il mergesort. In particolare, mostrare i "run" (cioè le porzioni di file ordinate durante prima passata) memorizzati su disco e i buffer in memoria centrale, evidenziando per ciascun buffer il record corrente. Mostrare anche i record prodotti dalle prime cinque chiamate di `next()`.

Run su disco		Buffer	Record prodotti dalle prime 5 <code>next()</code>
622	...		
201	...		
521	...		
322	...		
101	...		
421	...		
742	...		
871	...		
783	...		
144	...		
256	...		
585	...		
325	...		
435	...		
686	...		
885	...		
735	...		
386	...		
539	...		
178	...		
487	...		
839	...		
267	...		
647	...		
535	...		
171	...		
484	...		
838	...		
262	...		
646	...		

Basi di dati II — 8 aprile 2019 — Compito A

Domanda 2 (15%)

Considerare ancora quanto illustrato nella domanda precedente, con l'unica differenza nel numero di buffer disponibili: supporre che siano 3. Illustrare schematicamente i vari passi e i risultati intermedi (mostrando una quantità di dati che permetta di comprendere l'esecuzione dell'algoritmo).

622	...
201	...
521	...
322	...
101	...
421	...
742	...
871	...
783	...
144	...
256	...
585	...
325	...
435	...
686	...
885	...
735	...
386	...
539	...
178	...
487	...
839	...
267	...
647	...
535	...
171	...
484	...
838	...
262	...
646	...

Domanda 3 (20%)

Si considerino un sistema con blocchi di dimensione $B = 1000$ byte e una relazione $R(\underline{A}, B, \dots)$ di cardinalità pari circa a $L = 1.000.000$, con ennuple di $e = 100$ byte e campo chiave A di tipo intero e campo B di tipo stringa (ad esempio un nome). Supporre che il sistema offra

- le seguenti strutture primarie
 - struttura disordinata
 - struttura ordinata (rispetto ad un indice denso di tipo B-tree), ma con ordinamento non mantenuto (eventuali inserimenti vengono effettuati in coda al file)
 - struttura hash (dinamica) su un attributo (non necessariamente chiave)
- indici di tipo B-tree

Considerare un carico applicativo che preveda una pausa notturna (utilizzabile per riorganizzare le strutture fisiche) e le seguenti operazioni

1. inserimento di una ennupla (con verifica del vincolo di chiave), con frequenza giornaliera $f_1 = 100$;
2. ricerca di una ennupla sulla base del valore della chiave A , con frequenza giornaliera $f_2 = 1000$;
3. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) dell'attributo B , con frequenza giornaliera $f_3 = 10.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $s = 10$ ennuple.

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 abbiano profondità $p = 4$, (2) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice, (3) l'hash dinamico, per gli accessi puntuali, abbia costo unitario, (4) lettura e scrittura abbiano lo stesso costo. Inoltre, qualora si scelga una struttura ordinata, si assuma che la riorganizzazione sia eseguita tutte le sere e considerare un numero medio di record disordinati in coda pari a metà degli inserimenti di una giornata. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica. Non è necessario considerare il costo della eventuale riorganizzazione.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. struttura			
Costo Op. 1			
Costo Op. 2			
Costo Op. 3			
Costo tot			

Basi di dati II — 8 aprile 2019 — Compito A

Domanda 4 (20%)

Considerare ancora il caso illustrato nella domanda precedente, ma con riferimento ad una realtà che non preveda pause e quindi non permetta riorganizzazioni periodiche delle strutture e con le seguenti frequenze per le operazioni:

1. $f_1 = 100$
2. $f_2 = 10.000$
3. $f_3 = 1000$

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

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. struttura			
Costo Op. 1			
Costo Op. 2			
Costo Op. 3			
Costo tot			

Basi di dati II — 8 aprile 2019 — Compito A

Domanda 5 (25%)

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 $t_B = 10$ μ sec

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 $m = 10.000$ 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?

Basi di dati II

Prova parziale — 8 aprile 2019 — Compito B

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ Nome _____ Matricola _____

Basi di dati II — 8 aprile 2019 — Compito B

Domanda 1 (20%)

Considerare la relazione sotto schematizzata, definita su vari attributi, uno dei quali è la chiave, i cui valori sono mostrati. Supporre che la relazione abbia un fattore di blocco pari a 2 (e quindi occupi 15 blocchi) e che siano disponibili 8 buffer. Considerare l'esecuzione di un mergesort a più vie (e due passate) sulla relazione e mostrare lo stato delle strutture in memoria centrale e secondaria dopo l'esecuzione di cinque chiamate al metodo `next()` sullo scan che implementa il mergesort. In particolare, mostrare i "run" (cioè le porzioni di file ordinate durante prima passata) memorizzati su disco e i buffer in memoria centrale, evidenziando per ciascun buffer il record corrente. Mostrare anche i record prodotti dalle prime cinque chiamate di `next()`.

	Run su disco	Buffer	Record prodotti dalle prime 5 <code>next()</code>
	612 ...		
	211 ...		
	511 ...		
	312 ...		
	111 ...		
	411 ...		
	742 ...		
	871 ...		
	783 ...		
	144 ...		
	256 ...		
	585 ...		
	325 ...		
	435 ...		
	686 ...		
	885 ...		
	735 ...		
	386 ...		
	539 ...		
	178 ...		
	487 ...		
	839 ...		
	267 ...		
	647 ...		
	535 ...		
	171 ...		
	484 ...		
	838 ...		
	262 ...		
	646 ...		

Basi di dati II — 8 aprile 2019 — Compito B

Domanda 2 (15%)

Considerare ancora quanto illustrato nella domanda precedente, con l'unica differenza nel numero di buffer disponibili: supporre che siano 3. Illustrare schematicamente i vari passi e i risultati intermedi (mostrando una quantità di dati che permetta di comprendere l'esecuzione dell'algoritmo).

612	...
211	...
511	...
312	...
111	...
411	...
742	...
871	...
783	...
144	...
256	...
585	...
325	...
435	...
686	...
885	...
735	...
386	...
539	...
178	...
487	...
839	...
267	...
647	...
535	...
171	...
484	...
838	...
262	...
646	...

Basi di dati II — 8 aprile 2019 — Compito B

Domanda 3 (20%)

Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione $R(\underline{A}, B, \dots)$ di cardinalità pari circa a $N = 4.000.000$, con ennuple di $e = 100$ byte e campo chiave A di tipo intero e campo B di tipo stringa (ad esempio un nome). Supporre che il sistema offra

- le seguenti strutture primarie
 - struttura disordinata
 - struttura ordinata (rispetto ad un indice denso di tipo B-tree), ma con ordinamento non mantenuto (eventuali inserimenti vengono effettuati in coda al file)
 - struttura hash (dinamica) su un attributo (non necessariamente chiave)
- indici di tipo B-tree

Considerare un carico applicativo che preveda una pausa notturna (utilizzabile per riorganizzare le strutture fisiche) e le seguenti operazioni

1. inserimento di una ennupla (con verifica del vincolo di chiave), con frequenza giornaliera $f_1 = 100$;
2. ricerca di una ennupla sulla base del valore della chiave A , con frequenza giornaliera $f_2 = 1000$;
3. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) dell'attributo B , con frequenza giornaliera $f_3 = 10.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $n = 10$ ennuple.

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 abbiano profondità $p = 4$, (2) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice, (3) l'hash dinamico, per gli accessi puntuali, abbia costo unitario, (4) lettura e scrittura abbiano lo stesso costo. Inoltre, qualora si scelga una struttura ordinata, si assuma che la riorganizzazione sia eseguita tutte le sere e considerare un numero medio di record disordinati in coda pari a metà degli inserimenti di una giornata. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica. Non è necessario considerare il costo della eventuale riorganizzazione.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. struttura			
Costo Op. 1			
Costo Op. 2			
Costo Op. 3			
Costo tot			

Basi di dati II — 8 aprile 2019 — Compito B

Domanda 4 (20%)

Considerare ancora il caso illustrato nella domanda precedente, ma con riferimento ad una realtà che non preveda pause e quindi non permetta riorganizzazioni periodiche delle strutture e con le seguenti frequenze per le operazioni:

1. $f_1 = 100$
2. $f_2 = 10.000$
3. $f_3 = 1000$

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

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. struttura			
Costo Op. 1			
Costo Op. 2			
Costo Op. 3			
Costo tot			

Basi di dati II — 8 aprile 2019 — Compito B

Domanda 5 (25%)

Considerare un sistema con dischi con le seguenti caratteristiche

- tempo medio di posizionamento della testina (tempo di seek) $t_S = 5$ msec
- tempo medio di latenza (attesa dovuta alla rotazione) $t_L = 3$ msec
- tempo minimo di lettura di un blocco $t_B = 10$ μ sec

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 $m = 20.000$ 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?

Basi di dati II

Prova parziale — 8 aprile 2019 — Compito **A**

Cenni sulle soluzioni

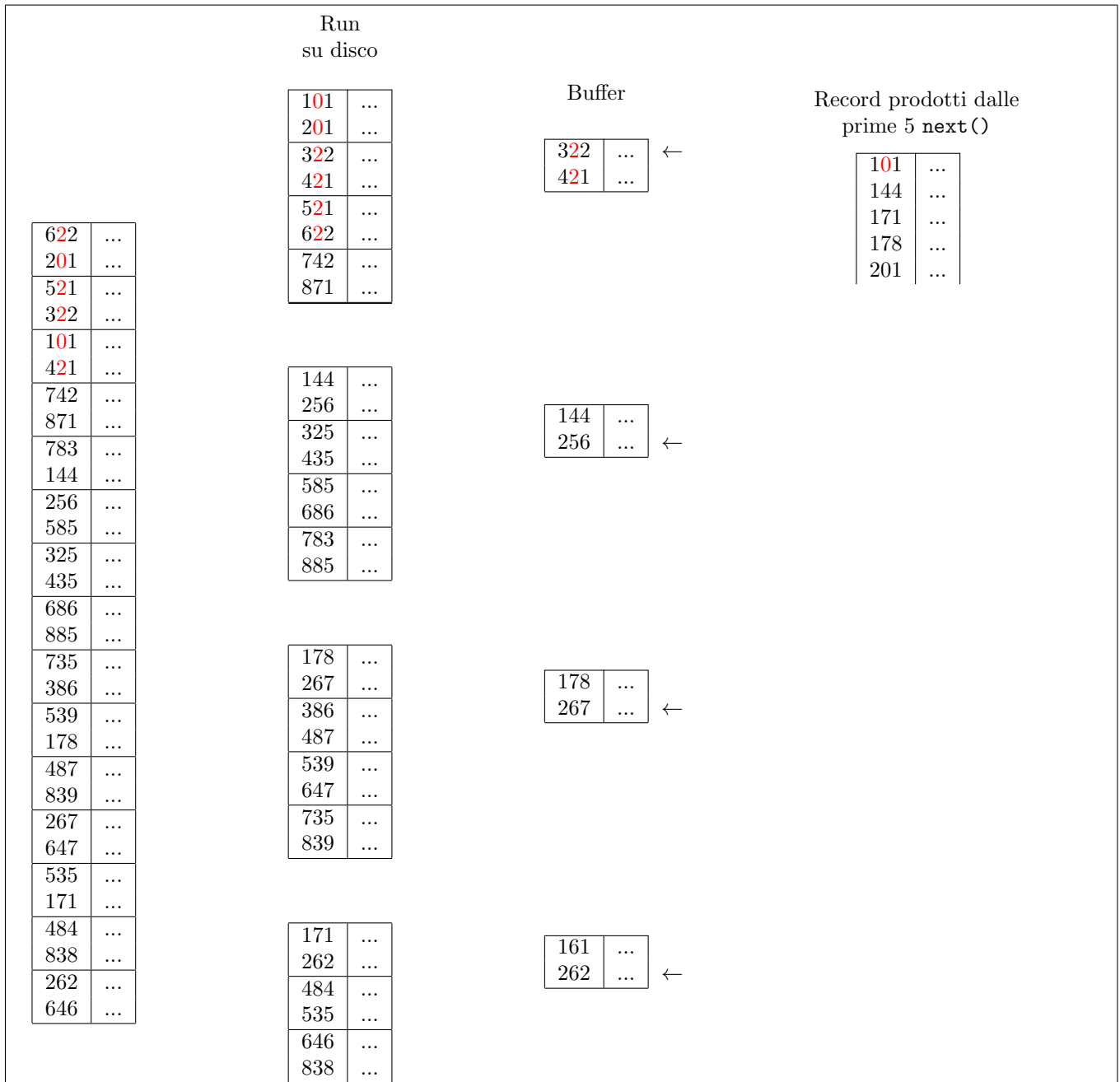
(per il compito A, gli altri sono simili, le varianti del testo sono in **rosso**)

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%)

Considerare la relazione sotto schematizzata, definita su vari attributi, uno dei quali è la chiave, i cui valori sono mostrati. Supporre che la relazione abbia un fattore di blocco pari a 2 (e quindi occupi 15 blocchi) e che siano disponibili 8 buffer. Considerare l'esecuzione di un mergesort a più vie (e due passate) sulla relazione e mostrare lo stato delle strutture in memoria centrale e secondaria dopo l'esecuzione di cinque chiamate al metodo `next()` sullo scan che implementa il mergesort. In particolare, mostrare i "run" (cioè le porzioni di file ordinate durante prima passata) memorizzati su disco e i buffer in memoria centrale, evidenziando per ciascun buffer il record corrente. Mostrare anche i record prodotti dalle prime cinque chiamate di `next()`.



Il numero ideale di buffer (da utilizzare sia nella prima sia nella seconda passata) è pari alla radice quadrata del numero dei blocchi e quindi a quattro.

Nella prima passata, si costruiscono quindi quattro run ordinati, ciascuno a partire da quattro blocchi del file originario. L'ordinamento di ciascun run può essere effettuato in memoria centrale, usando quattro buffer. Poiché il risultato della prima passata viene materializzato, vengono mostrati i run ordinati.

Nella seconda passata, si carica un blocco per ciascuno dei run e si fa il merge usando nuovamente quattro buffer. La seconda passata viene svolta in pipeline, e quindi i buffer sono mostrati con il contenuto che hanno quando è stato appena prodotto il quinto record.

Domanda 2 (15%)

Considerare ancora quanto illustrato nella domanda precedente, con l'unica differenza nel numero di buffer disponibili: supporre che siano 3. Illustrare schematicamente i vari passi e i risultati intermedi (mostrando una quantità di dati che permetta di comprendere l'esecuzione dell'algoritmo).

622	...
201	...
521	...
322	...
101	...
421	...
742	...
871	...
783	...
144	...
256	...
585	...
325	...
435	...
686	...
885	...
735	...
386	...
539	...
178	...
487	...
839	...
267	...
647	...
535	...
171	...
484	...
838	...
262	...
646	...

Con tre buffer, due passate non sono sufficienti, ma ne servono di più. Piccole varianti sono ammissibili. La più semplice prevede di usare due buffer più uno per i risultati e quindi quattro passate: nella prima si formano run di due blocchi, nella seconda si fondono in run di quattro, nella terza in run di otto e nella quarta si produce il risultato.

In alternativa, si possono formare run di tre blocchi nella prima passata, poi fonderli in run di sei, poi di dodici e poi si genera il risultato.

Ci si aspettava una schematizzazione dei risultati intermedi (con i blocchi dopo ciascuna fase)

Domanda 3 (20%)

Si considerino un sistema con blocchi di dimensione $B = 1000$ byte e una relazione $R(\underline{A}, B, \dots)$ di cardinalità pari circa a $L = 1.000.000$, con ennuple di $e = 100$ byte e campo chiave A di tipo intero e campo B di tipo stringa (ad esempio un nome). Supporre che il sistema offra

- le seguenti strutture primarie
 - struttura disordinata
 - struttura ordinata (rispetto ad un indice denso di tipo B-tree), ma con ordinamento non mantenuto (eventuali inserimenti vengono effettuati in coda al file)
 - struttura hash (dinamica) su un attributo (non necessariamente chiave)
- indici di tipo B-tree

Considerare un carico applicativo che preveda una pausa notturna (utilizzabile per riorganizzare le strutture fisiche) e le seguenti operazioni

1. inserimento di una ennupla (con verifica del vincolo di chiave), con frequenza giornaliera $f_1 = 100$;
2. ricerca di una ennupla sulla base del valore della chiave A , con frequenza giornaliera $f_2 = 1000$;
3. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) dell'attributo B , con frequenza giornaliera $f_3 = 10.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $s = 10$ ennuple.

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 abbiano profondità $p = 4$, (2) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice, (3) l'hash dinamico, per gli accessi puntuali, abbia costo unitario, (4) lettura e scrittura abbiano lo stesso costo. Inoltre, qualora si scelga una struttura ordinata, si assuma che la riorganizzazione sia eseguita tutte le sere e considerare un numero medio di record disordinati in coda pari a metà degli inserimenti di una giornata. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica. Non è necessario considerare il costo della eventuale riorganizzazione.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. strutt.	Struttura ordinata con indice su B e indice secondario su A	Struttura hash su A e indice secondario su B	Struttura disordinata con indice secondario su A e su B
Costo Op. 1	$p-2+1+p-2+1+2=ca.8$: vanno visitati $(p-2)$ e aggiornati (1) entrambi gli indici e va letto e scritto il blocco per il nuovo record	$2+p-2+1= ca. 5$	$p-2+1+p-2+1+2= ca. 9$
Costo Op. 2	$p-2+1 = 3$	1	$p-2+1 = 3$
Costo Op. 3	$p-2+1 = 3$; nella maggior parte dei casi i 10 record sono nello stesso blocco. Gli aggiornamenti influiscono pochissimo (100 su un milione)	$p-2+10 = 12$; i record sono in blocchi diversi	$p-2+10 = 12$; i record sono in blocchi diversi
Tot	$800 + 3000 + 30.000 = ca 34.000$	$500+1000+120.000 = ca 120.000$	$900+3000+120.000 = ca 124.000$

Basi di dati II — 8 aprile 2019 — Compito A

Domanda 4 (20%)

Considerare ancora il caso illustrato nella domanda precedente, ma con riferimento ad una realtà che non preveda pause e quindi non permetta riorganizzazioni periodiche delle strutture e con le seguenti frequenze per le operazioni:

1. $f_1 = 100$
2. $f_2 = 10.000$
3. $f_3 = 1000$

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 con indice su B e indice secondario su A (ha poco senso perché diventa presto disordinata e quindi la ignoriamo)	Struttura hash su A e indice secondario su B	Struttura disordinata con indice secondario su A e su B
Costo Op. 1		$2+p-2+1 = \text{ca. } 5$	$p-2+1+p-2+1+2 = \text{ca. } 8$
Costo Op. 2		1	$p-2+1 = 3$
Costo Op. 3		$p-2+10 = 12$; i record sono in blocchi diversi	$p-2+10 = 12$; i record sono in blocchi diversi
Tot		$500+10.000+12.000 = \text{ca } 22.000$	$800+30.000+12.000 = \text{ca } 43.000$

Domanda 5 (25%)

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 $t_B = 10$ μ sec

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

$$t_{tot} = t_S + t_L + t_B = 4 + 4 + 0,01 \text{ msec} = \text{ca } 8 \text{ msec}$$

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

$$t_{tot} + (F - 1) \times t_B = \text{ca } 10 \text{ msec}$$

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:

$$((p + 1) + 3 \times ((p - 1) + 1)) \times t_{tot} = \text{ca } 135 \text{ msec}$$

4. Qual è il tempo che si può ipotizzare necessario per eseguire $m = 10.000$ 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, due accessi 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):

$$(p - 3 + 1) \times m \times t_{tot} = \text{ca } 160 \text{ sec}$$