Basi di dati II

Prova parziale — 23 maggio 2016 — Compito A

Tempo a disposizione: un'ora e trenta minuti.

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

Domanda 1 (20%) Considerare un sistema con dischi con $T = 400$ blocchi per traccia e con

- 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 (N.B. non servono calcolatrici, i calcoli sono semplici; **se si ritengono le informazioni imprecise, dare risposte approssimate, usando il buon senso**).

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 $N = 200$ blocchi contigui, non letti di recente?

3. Qual è il tempo che si può ipotizzare necessario per eseguire un accesso diretto ad un record di un file attraverso un indice che abbia profondità $p_I = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 100$ e che sia stato usato di recente, ma in modo non molto intenso?

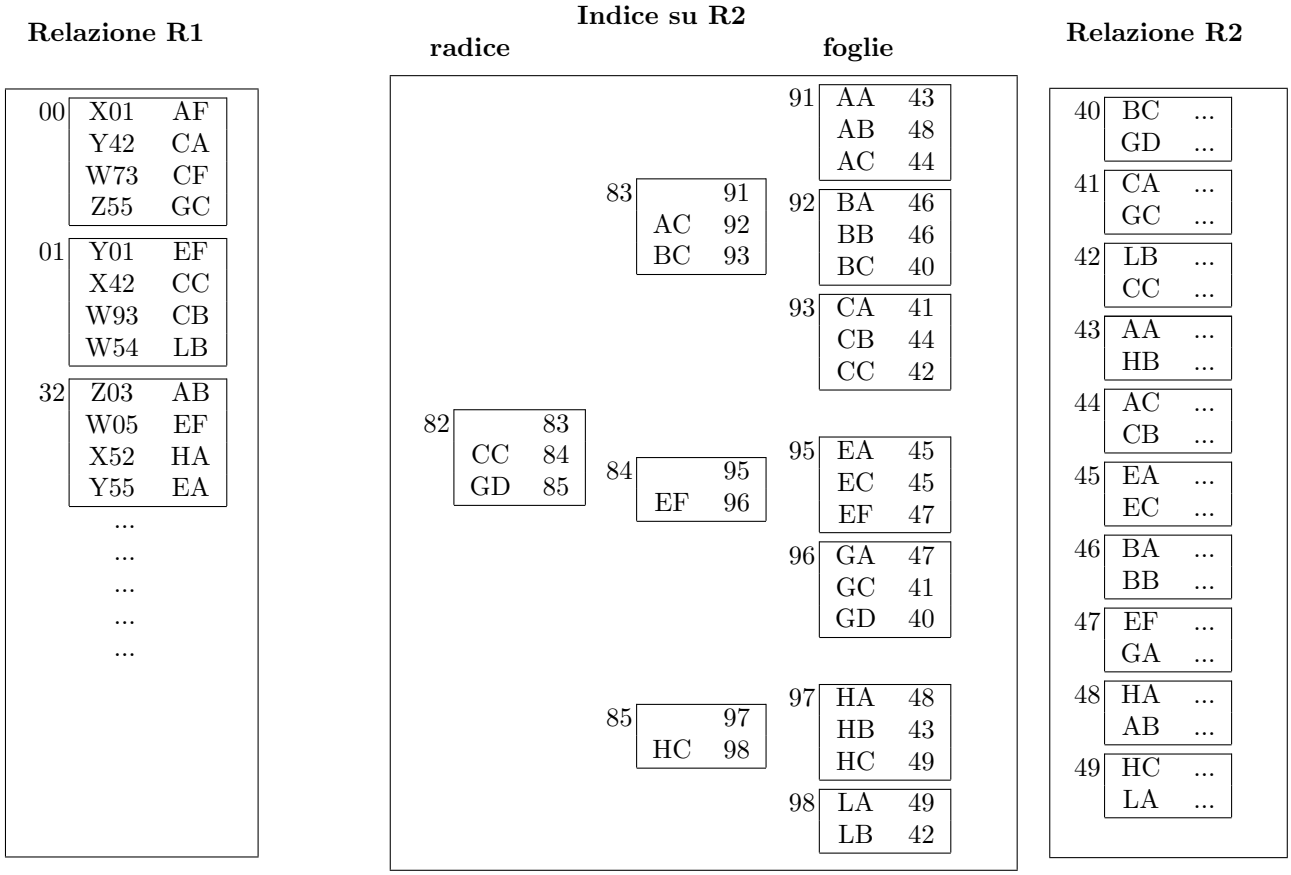
4. Qual è il tempo che si può ipotizzare necessario per eseguire $n = 1000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p_I = 4$, fan-out $f_I = 100$, con disponibilità di circa $P = 150$ pagine di buffer?

Domanda 2 (20%)
Si consideri un B-tree con nodi intermedi che contengono tre chiavi e quattro puntatori e foglie con tre chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 41, 57, 11, 32, 20, 27, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di quattro chiavi, di otto chiavi e alla fine.

Mostrare poi l'albero dopo l'eliminazione della chiave 20 dall'ultimo albero ottenuto in risposta alla domanda precedente.

Basi di dati II — 23 maggio 2016 — Compito A

Domanda 3 (25%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **quattro** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

Basi di dati II — 23 maggio 2016 — Compito A

Domanda 4 (35%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres, si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where B=21`
		Merge join

Motivare ciò, valutando, per ciascuna delle due interrogazioni, il costo di un piano di esecuzione con hash join e uno con merge join (e che, per l'operazione 2, includa la selezione), supponendo che

- le relazioni abbiano L_1=1.000.000 ed L_2=2.000.000 ennuple, (con fattore di blocco f_1=10 e f_2=20)
- l'attributo B in R1 abbia circa 100.000 valori diversi
- entrambi gli indici abbiano i=4 livelli (radice e foglie incluse) e fattore di blocco massimo f_i=100
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=500.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico.

	Hash join	Merge join
1.		
2.		

Indicare come cambiano i costi, per l'operazione 2, se sull'attributo B è definito un indice e se l'interrogazione richiede solo gli attributi A, B, C (quindi nessun attributo di R2). Mostrare un'unica risposta che tenga conto di entrambe le varianti.

	Hash join	Merge join
2bis.		

Basi di dati II

Prova parziale — 23 maggio 2016 — Compito B

Tempo a disposizione: un'ora e trenta minuti.

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

Domanda 1 (20%) Considerare un sistema con dischi con $T = 400$ blocchi per traccia e con

- tempo medio di posizionamento della testina (tempo di seek) $t_S = 5$ msec
- tempo medio di latenza (attesa dovuta alla rotazione) $t_L = 2$ msec
- tempo minimo di lettura di un blocco $t_B = 10$ μsec

Rispondere alle seguenti domande mostrando formula e valore numerico (N.B. non servono calcolatrici, i calcoli sono semplici; **se si ritengono le informazioni imprecise, dare risposte approssimate, usando il buon senso**).

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 $N = 200$ blocchi contigui, non letti di recente?

3. Qual è il tempo che si può ipotizzare necessario per eseguire un accesso diretto ad un record di un file attraverso un indice che abbia profondità $p_I = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 100$ e che sia stato usato di recente, ma in modo non molto intenso?

4. Qual è il tempo che si può ipotizzare necessario per eseguire $n = 2000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p_I = 4$, fan-out $f_I = 100$, con disponibilità di circa $P = 150$ pagine di buffer?

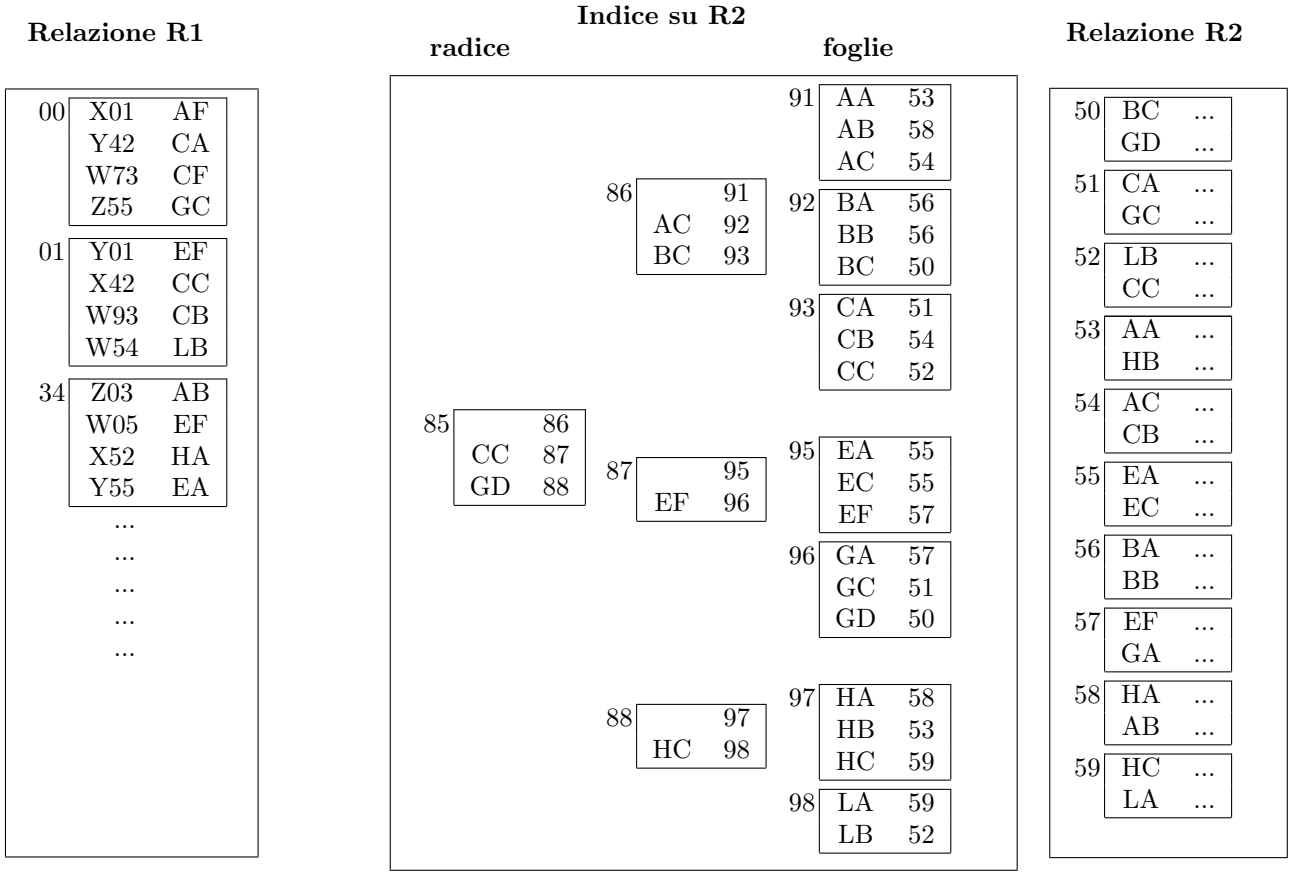
Basi di dati II — 23 maggio 2016 — Compito B

Domanda 2 (20%)
Si consideri un B-tree con nodi intermedi che contengono tre chiavi e quattro puntatori e foglie con tre chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 43, 51, 10, 32, 21, 25, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di quattro chiavi, di otto chiavi e alla fine.

Mostrare poi l'albero dopo l'eliminazione della chiave 21 dall'ultimo albero ottenuto in risposta alla domanda precedente.

Basi di dati II — 23 maggio 2016 — Compito B

Domanda 3 (25%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **quattro** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

Basi di dati II — 23 maggio 2016 — Compito B

Domanda 4 (35%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria)

T1(A,B,C), T2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres, si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from T1 join T2 on C=D`	Hash join

2.	`select *`
	`from T1 join T2 on C=D`
	`where B=21`
		Merge join

Motivare ciò, valutando, per ciascuna delle due interrogazioni, il costo di un piano di esecuzione con hash join e uno con merge join (e che, per l'operazione 2, includa la selezione), supponendo che

- le relazioni abbiano N_1=2.000.000 ed N_2=1.000.000 ennuple, (con fattore di blocco f_1=20 e f_2=10)
- l'attributo B in R1 abbia circa 100.000 valori diversi
- entrambi gli indici abbiano p=4 livelli (radice e foglie incluse) e fattore di blocco massimo f_i=100
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=500.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico.

	Hash join	Merge join
1.		
2.		

Indicare come cambiano i costi, per l'operazione 2, se sull'attributo B è definito un indice e se l'interrogazione richiede solo gli attributi A, B, C (quindi nessun attributo di R2). Mostrare un'unica risposta che tenga conto di entrambe le varianti.

	Hash join	Merge join
2bis.		

Basi di dati II

Prova parziale — 23 maggio 2016 — Compito A

Cenni sulle soluzioni

Tempo a disposizione: un'ora e trenta minuti.

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

Domanda 1 (20%) Considerare un sistema con dischi con $T = 400$ blocchi per traccia e con

- 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 (N.B. non servono calcolatrici, i calcoli sono semplici; **se si ritengono le informazioni imprecise, dare risposte approssimate, usando il buon senso**).

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

$$t_{tot} = t_S + t_L + t_B = 4 + 3 + 0,015 \text{ msec } = \text{ ca } 7 \text{ msec}$$

2. Qual è il tempo medio necessario per la scansione sequenziale di un file costituito da $N = 200$ 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

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

3. Qual è il tempo che si può ipotizzare necessario per eseguire un accesso diretto ad un record di un file attraverso un indice che abbia profondità $p_I = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 100$ e che sia stato usato di recente, ma in modo non molto intenso?

Si può immaginare che al massimo la radice dell'indice si trovi nel buffer, ma non gli altri nodi. Quindi, tre accessi per l'indice e uno per il record del file, in posizioni non prevedibili:

$$4 \times t_{tot} = \text{ ca } 28 \text{ msec}$$

4. Qual è il tempo che si può ipotizzare necessario per eseguire $n = 1000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p_I = 4$, fan-out $f_I = 100$, con disponibilità di circa $P = 150$ pagine di buffer?

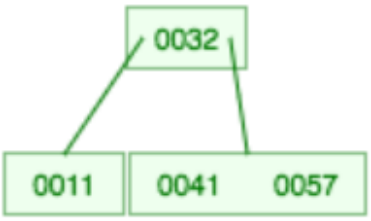
Si può immaginare che la radice e un altro livello dell'indice restino nel buffer, dopo il primo caricamento, quindi, per ogni record, due accessi all'indice e uno al file, in posizioni non prevedibili (qualche accesso in più per il caricamento iniziale e qualcuno in meno per blocchi acceduti più volte):

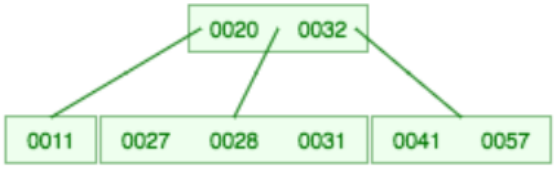
$$3 \times n \times t_{tot} = \text{ ca } 21 \text{ sec}$$

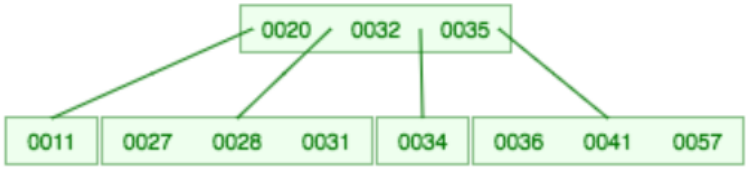
Basi di dati II — 23 maggio 2016 — Compito A

Domanda 2 (20%)
Si consideri un B-tree con nodi intermedi che contengono tre chiavi e quattro puntatori e foglie con tre chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 41, 57, 11, 32, 20, 27, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di quattro chiavi, di otto chiavi e alla fine.

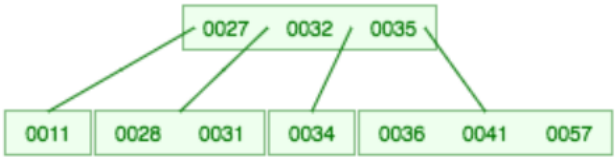
Vengono mostrate le soluzioni per il compito A. Le altre sono analoghe (isomorfe).





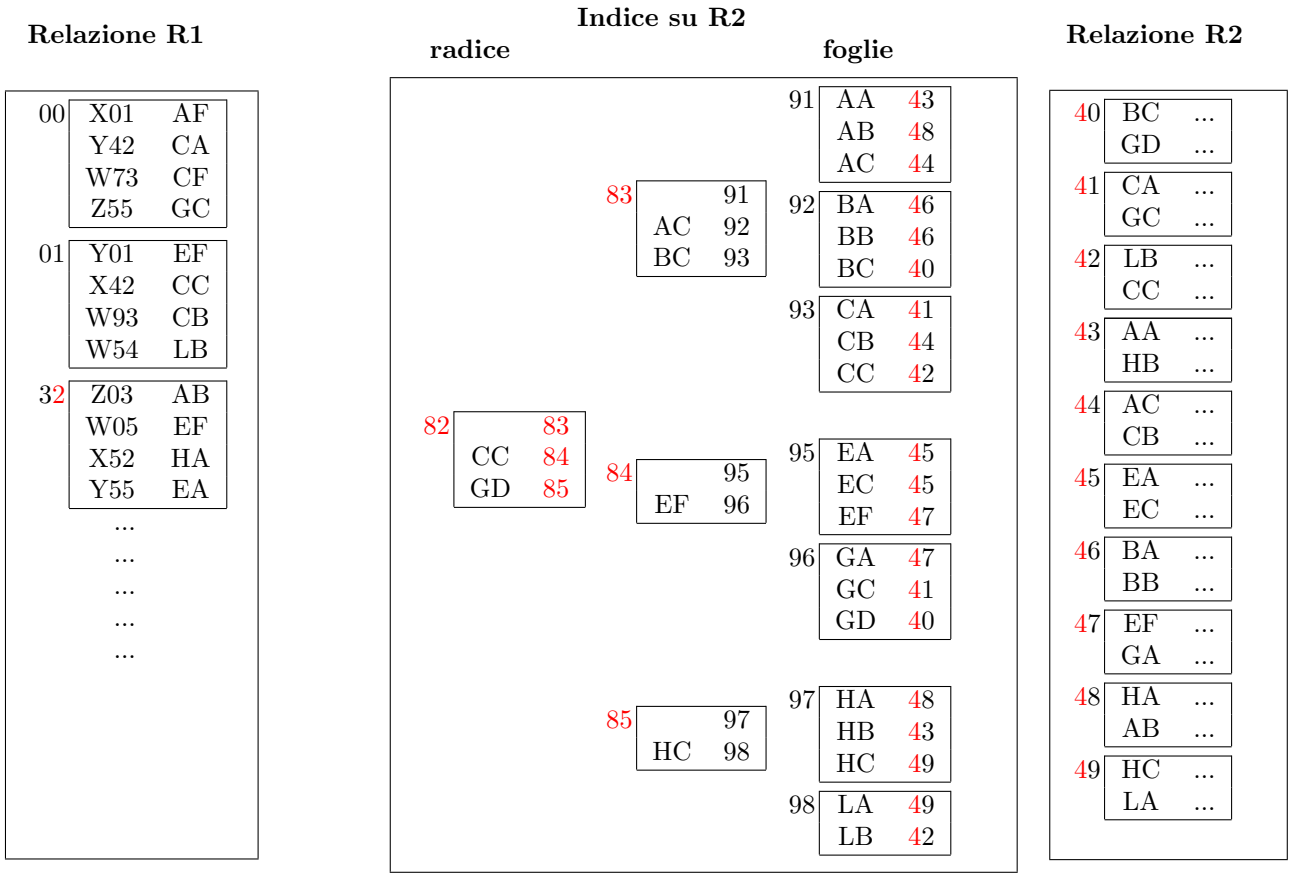


Mostrare poi l'albero dopo l'eliminazione della chiave 20 dall'ultimo albero ottenuto in risposta alla domanda precedente.



Basi di dati II — 23 maggio 2016 — Compito A

Domanda 3 (25%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **quattro** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

00, 82, 83, 92,
82, 83, 93, 41,
82, 84, 95,
82, 84, 96, 41,
01, 82, 84, 95, 47

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

00, 82, 83, 92,
93, 41,
82, 84, 95,
96, 41,
01, 82, 84, 95, 47

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

01, 95, 84, 47

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

00, 82, 83, 92,
93, 41, 82,
84, 95, 96,
41, 01, 82,
84, 95, 47

Basi di dati II — 23 maggio 2016 — Compito A

Domanda 4 (35%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres, si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where B=21`
		Merge join

Motivare ciò, valutando, per ciascuna delle due interrogazioni, il costo di un piano di esecuzione con hash join e uno con merge join (e che, per l'operazione 2, includa la selezione), supponendo che

- le relazioni abbiano L_1=1.000.000 ed L_2=2.000.000 ennuple, (con fattore di blocco f_1=10 e f_2=20)
- l'attributo B in R1 abbia circa 100.000 valori diversi
- entrambi gli indici abbiano i=4 livelli (radice e foglie incluse) e fattore di blocco massimo f_i=100
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=500.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico.

	Hash join	Merge join
1.	Con i buffer disponibili (che sono in numero maggiore della radice quadrata del numero di blocchi del più piccolo dei due file), l'hash-join si può eseguire in due passate, con un costo: $3 \times (\frac{L_1}{f_1} + \frac{L_2}{f_2}) = 600.000$	Come ordine di grandezza, il costo dovrebbe essere paragonabile a quello dell'hash join, però per eseguire i due merge in parallelo potrebbero servire più buffer. Usando l'indice si potrebbe forse procedere meglio, ma non in modo rilevante. Su relazioni di medie dimensioni, Postgres presenta un costo per il mergejoin leggermente superiore a quello dell'hash join, anche se paragonabile
2.	La selezione su B produce un valore e (vista l'ipotesi sul numero di valori diversi) possiamo supporre che produca $b = 10$ ennuple. La selezione richiede una scansione della prima relazione. L'hash join può venire poi eseguito con una singola scansione della seconda relazione: $(\frac{L_1}{f_1} + \frac{L_2}{f_2}) = 200.000$	R2 va letta per intero per la selezione e il risultato può essere ordinato (in memoria) su C. Il merge join va un scansione dell'indice, visitando tutte le foglie. Costo: $(\frac{L_1}{f_1} + \frac{L_2}{f_i}) =$ ca.120.000 Con i numeri dell'esercizio, il nested loop con indice su R2 costa probabilmente di meno: $(\frac{L_1}{f_1} + b \times i) =$ ca.100.000

Indicare come cambiano i costi, per l'operazione 2, se sull'attributo B è definito un indice e se l'interrogazione richiede solo gli attributi A, B, C (quindi nessun attributo di R2). Mostrare un'unica risposta che tenga conto di entrambe le varianti.

	Hash join	Merge join
2bis.	Accesso diretto per la selezione e poi scansione di R2. Costo: $((i + b) + \frac{L_2}{f_2}) =$ ca.100.000	Accesso diretto per la selezione e poi scansione dell'indice di R2. Costo: $((i + b) + \frac{L_2}{f_i}) =$ ca.20.000 In questo caso il nested loop con indice converrebbe di sicuro: $((i + b) + b \times i) =$ ca.50