Tecnologie e architetture per la gestione dei dati

Prova parziale — 12 aprile 2023

Tempo a disposizione: un'ora e 10 minuti.

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

Tecnologie e architetture per la gestione dei dati — 12 aprile 2023

Domanda 1 (30%)
Si considerino un sistema con blocchi di $B = 4000$ byte e una relazione *R(Matricola,CodiceFiscale,Cognome,....)* di cardinalità pari circa a $L = 10.000.000$, con ennuple di $l = 40$ byte. Il campo *Matricola* è chiave di tipo intero e il campo *CodiceFiscale* è di tipo stringa e anch'esso chiave — quindi la relazione ha due chiavi, ognuna delle quali, da sola, identifica univocamente le ennuple. Supporre che il sistema offra

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

Considerare un carico applicativo con le seguenti operazioni

1. inserimento di una ennupla con numero di matricola maggiore di tutti quelli già presenti (succede sempre così ma viene comunque verificato il vincolo); anche il codice fiscale è diverso da quelli esistenti (ma anche in questo caso il vincolo va verificato), con frequenza giornaliera $f_1 = 1000$
2. ricerca di una ennupla sulla base del valore (completo) della matricola, con frequenza giornaliera $f_2 = 1000$;
3. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) del codice fiscale, con frequenza giornaliera $f_3 = 10.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $s_3 = 3$ ennuple;
4. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) del cognome, con frequenza giornaliera $f_4 = 1$; supporre che il valore parziale sia poco selettivo e porti alla identificazione, in media, di $s_4 = 1000$ ennuple;
5. scansione dell'intera relazione, ordinata sulla base del valore della matricola, con frequenza giornaliera $f_5 = 4$;

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.
Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere commentando prima di tutto, qui sotto, i criteri sulla base dei quali sono state scelte le alternative ritenute migliori e poi alla pagina successiva specificare i costi, in forma sia simbolica sia numerica.

Tecnologie e architetture per la gestione dei dati — 12 aprile 2023

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descrizione struttura			
Costo operazione 1			
Costo operazione 2			
Costo operazione 3			
Costo operazione 4			
Costo operazione 5			
Costo totale			

Tecnologie e architetture per la gestione dei dati — 12 aprile 2023

Domanda 2 (25%) Come noto, in Postgres (e, con sintassi diverse, negli altri sistemi) è possibile ordinare fisicamente una relazione con il comando `CLUSTER`. L'ordinamento viene specificato sulla base di un indice già definito sulla stessa relazione. L'ordinamento **non viene però mantenuto** (se non eseguendo nuovamente il comando `CLUSTER`).

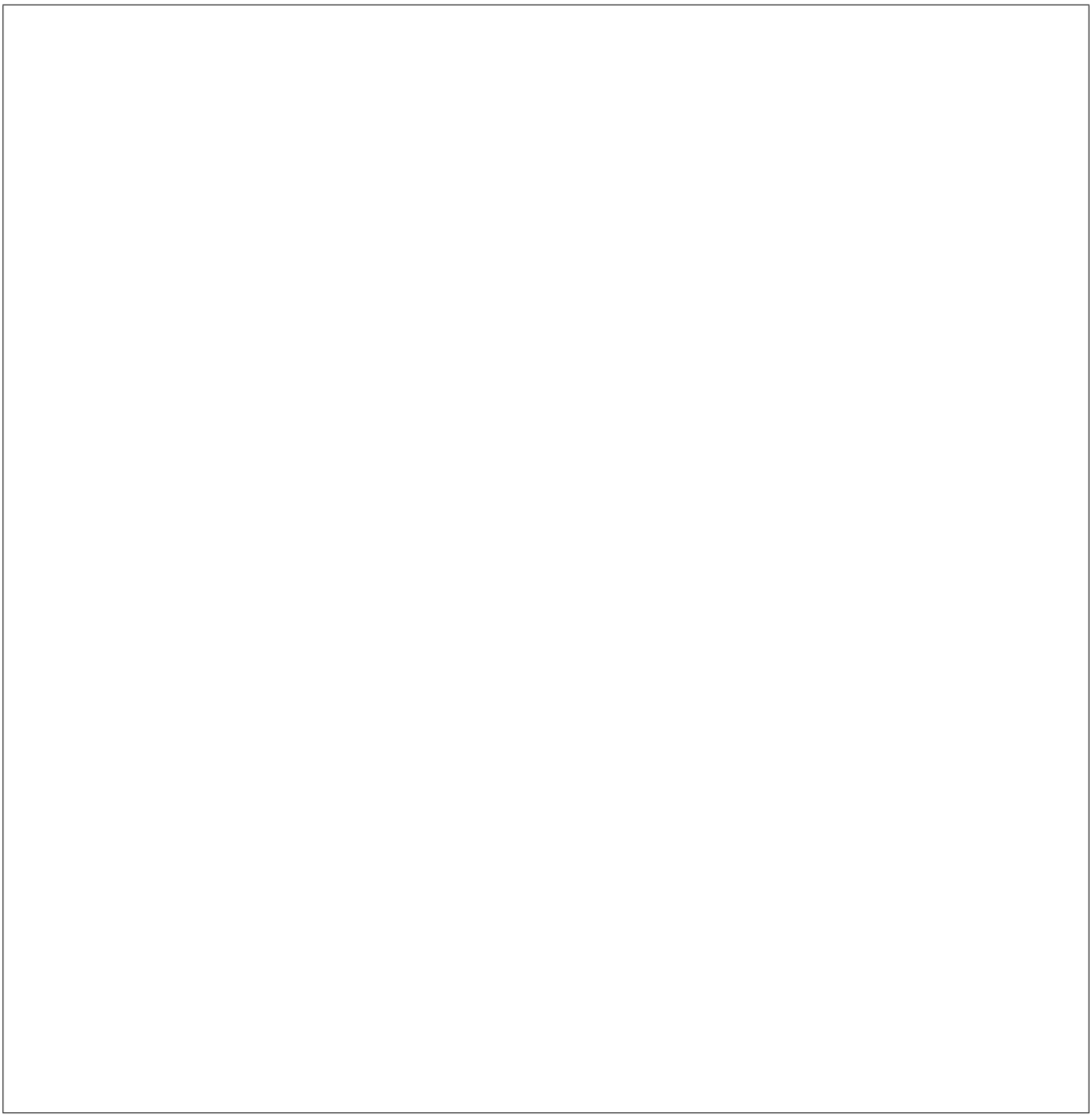
Sia data una relazione $R(\underline{A}, B, C)$ contenente circa $L = 4.000.000$ ennuple di $r = 40$ Byte ciascuna, con $c = 4000$ valori diversi per l'attributo C, uniformemente distribuiti (quindi per ogni valore di C ci sono circa $L/c = 1000$ ennuple con tale valore). Supporre che i blocchi abbiano dimensione $B = 4$KB approssimabile come 4.000. Considerare le operazioni sotto elencate e indicare, in ciascun riquadro sotto, il costo dell'operazione indicata subito prima del riquadro stesso, con brevi spiegazioni:

1. `CREATE INDEX RCIX ON R (C)` (creazione di un indice su C; supporre che abbia profondità $p = 3$)
2. `CLUSTER R USING RCIX` (ordinamento di R sulla base dell'indice e quindi sull'attributo C)
3. `SELECT * FROM R WHERE C = 100`

4. inserimento di $L/100 = 40.000$ ennuple, in ordine casuale, con valori di C pure uniformemente distribuiti (quindi si può supporre che vengano inserite 10 ennuple per ogni valore di C)
5. `SELECT * FROM R WHERE C = 100`

6. `CLUSTER` (questa operazione riordina la relazione; descrivere brevemente l'algoritmo utilizzato e il relativo costo; supporre una disponibilità di 1000 pagine di buffer)

Domanda 3 (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: 10, 9, 8, 7, 6, 13, 14, 15, 30, 32, 5. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.



Domanda 4 (25%)
Si consideri una base di dati con le relazioni

- R1(A,B,C,D) con
  - vincolo di integrità referenziale fra l'attributo D e la chiave E della relazione R2
  - N_1=2.000.000 ennuple e fattore di blocco f_1=200
  - b=200 valori diversi sull'attributo B (tutti gli interi compresi fra 1 e b)
  - c=200.000 valori diversi sull'attributo C (tutti gli interi compresi fra 1 e c)
  - una struttura disordinata, un indice sulla chiave primaria A e un altro sull'attributo C;
- R2(E,F,G) con
  - N_2=1.000.000 ennuple e fattore di blocco f_2=10
  - una struttura disordinata, un indice sulla chiave primaria E

Supponendo che:

- gli indici abbiano tutti p=4 livelli (contando anche radice e foglie) e fattore di blocco massimo f_i=100
- il sistema esegua i join come **nested loop** (eventualmente utilizzando indici) oppure come **hash join**
- le pagine di buffer disponibili siano circa q=1000,

valutare il costo (indicandolo in modo sia simbolico sia numerico) di ciascuna delle interrogazioni seguenti **indicando anche il metodo di join che viene utilizzato**:

```
select *
from R1 join R2 on D=E
```

```
select *
from R1 join R2 on D=E
where B=5
```

```
select *
from R1 join R2 on D=E
where C=502
```

Tecnologie e architetture per la gestione dei dati

Prova parziale — 12 aprile 2023

Cenni sulle soluzioni

Tempo a disposizione: un'ora e 10 minuti.

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

Tecnologie e architetture per la gestione dei dati — 12 aprile 2023

Domanda 1 (30%)
Si considerino un sistema con blocchi di $B = 4000$ byte e una relazione *R(Matricola,CodiceFiscale,Cognome,....)* di cardinalità pari circa a $L = 10.000.000$, con ennuple di $l = 40$ byte. Il campo *Matricola* è chiave di tipo intero e il campo *CodiceFiscale* è di tipo stringa e anch'esso chiave — quindi la relazione ha due chiavi, ognuna delle quali, da sola, identifica univocamente le ennuple. Supporre che il sistema offra

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

Considerare un carico applicativo con le seguenti operazioni

1. inserimento di una ennupla con numero di matricola maggiore di tutti quelli già presenti (succede sempre così ma viene comunque verificato il vincolo); anche il codice fiscale è diverso da quelli esistenti (ma anche in questo caso il vincolo va verificato), con frequenza giornaliera $f_1 = 1000$
2. ricerca di una ennupla sulla base del valore (completo) della matricola, con frequenza giornaliera $f_2 = 1000$;
3. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) del codice fiscale, con frequenza giornaliera $f_3 = 10.000$; supporre che il valore parziale sia abbastanza selettivo e porti alla identificazione, in media, di $s_3 = 3$ ennuple;
4. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) del cognome, con frequenza giornaliera $f_4 = 1$; supporre che il valore parziale sia poco selettivo e porti alla identificazione, in media, di $s_4 = 1000$ ennuple;
5. scansione dell'intera relazione, ordinata sulla base del valore della matricola, con frequenza giornaliera $f_5 = 4$;

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.
Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere commentando prima di tutto, qui sotto, i criteri sulla base dei quali sono state scelte le alternative ritenute migliori e poi alla pagina successiva specificare i costi, in forma sia simbolica sia numerica.

Osservazioni

- R ha un fattore di blocco B/l pari a 100 e quindi occupa $L_B = (L/(B/l)) = 100.000$ blocchi.
- L'ordinamento sulla matricola si mantiene facilmente (gli aggiornamenti vanno sempre in coda) ed è utile per l'op. 5 (altrimenti servirebbe un ordinamento di costo pari a $3 \times L_B = (L/(B/l)) = 300.000$.
- In alternativa si potrebbe pensare ad un ordinamento su Cognome, per favorire l'op.4, che però ha una frequenza molto bassa e quindi il beneficio sarebbe minimo; in tal caso sarebbe però necessario in qualche modo mantenere l'ordinamento — come? E comunque si penalizzerebbe l'operazione 5, il cui costo verrebbe triplicato
- Indice su matricola è utile per l' operazione 2
- Indici su codice fiscale è necessario per la verifica di chiave (altrimenti servirebbe una scansione sequenziale) e utile per l'operazione 3.
- L'indice sul cognome ha costi aggiuntivi per l'operazione 1 (aggiornamento dell'indice) e benefici per l'operazione 4, è necessario valutare quantitativamente pro e contro

Quindi, sono comunque opportuni

- ordinamento su matricola e indice primario
- indice su codice fiscale

Va valutato l'indice su cognome

Tecnologie e architetture per la gestione dei dati — 12 aprile 2023

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. strutt.	Struttura ordinata con indice primario su Matricola e indice secondario su CodiceFiscale	Idem più indice secondario su Cognome	
Costo Op. 1	visita e aggiornamento dei due indici $p_1 - 2 + 1$ e $p_2 - 2 + 1$ e lettura e scrittura dell'ultimo blocco totale 7 pesato $7 \times 1000 =$ **7000**	idem più visita e aggiornamento del terzo indice totale 10 pesato $10 \times 1000 =$ **10.000**	
Costo Op. 2	p_1-2+1=ca.2: visita indice (p_1-2) e 1 record; pesato **2000**	idem **2000**	
Costo Op. 3	p_2-2+3=ca.5: visita indice (p_2-2) e 3 record; totale 5, pesato **50.000**	idem **50.000**	
Costo Op. 4	Serve scansione: L/(B/c)=100.000 totale **100.000**	p_2-2+1000=ca.1000: visita indice (p_2-2) e 1000 record; totale ca 1000, pesato **1000**	
Costo Op. 5	Essendo il file ordinato, basta la scansione: L/(B/c)=100.000 totale **400.000**	Idem **400.000**	
Tot	**ca 560.000**	**ca 460.000**	

Tecnologie e architetture per la gestione dei dati — 12 aprile 2023

Domanda 2 (25%) Come noto, in Postgres (e, con sintassi diverse, negli altri sistemi) è possibile ordinare fisicamente una relazione con il comando `CLUSTER`. L'ordinamento viene specificato sulla base di un indice già definito sulla stessa relazione. L'ordinamento **non viene però mantenuto** (se non eseguendo nuovamente il comando `CLUSTER`).

Sia data una relazione $R(\underline{A}, B, C)$ contenente circa $L = 4.000.000$ ennuple di $r = 40$ Byte ciascuna, con $c = 4000$ valori diversi per l'attributo C, uniformemente distribuiti (quindi per ogni valore di C ci sono circa $L/c = 1000$ ennuple con tale valore). Supporre che i blocchi abbiano dimensione $B = 4$KB approssimabile come 4.000. Considerare le operazioni sotto elencate e indicare, in ciascun riquadro sotto, il costo dell'operazione indicata subito prima del riquadro stesso, con brevi spiegazioni:

1. `CREATE INDEX RCIX ON R (C)` (creazione di un indice su C; supporre che abbia profondità $p = 3$)
2. `CLUSTER R USING RCIX` (ordinamento di R sulla base dell'indice e quindi sull'attributo C)
3. `SELECT * FROM R WHERE C = 100`

I record con lo stesso valore di `C` sono $L/c = 1000$ e si trovano in blocchi consecutivi. Visto che il fattore di blocco è $B/r = 100$, i blocchi che li contengono sono $(L/c)/(B/r)$, quindi 10 o 11 e quindi serviranno 13 o 14 accessi ($p = 3$ per l'indice più quelli per i blocchi). L'indice si visita una volta sola.

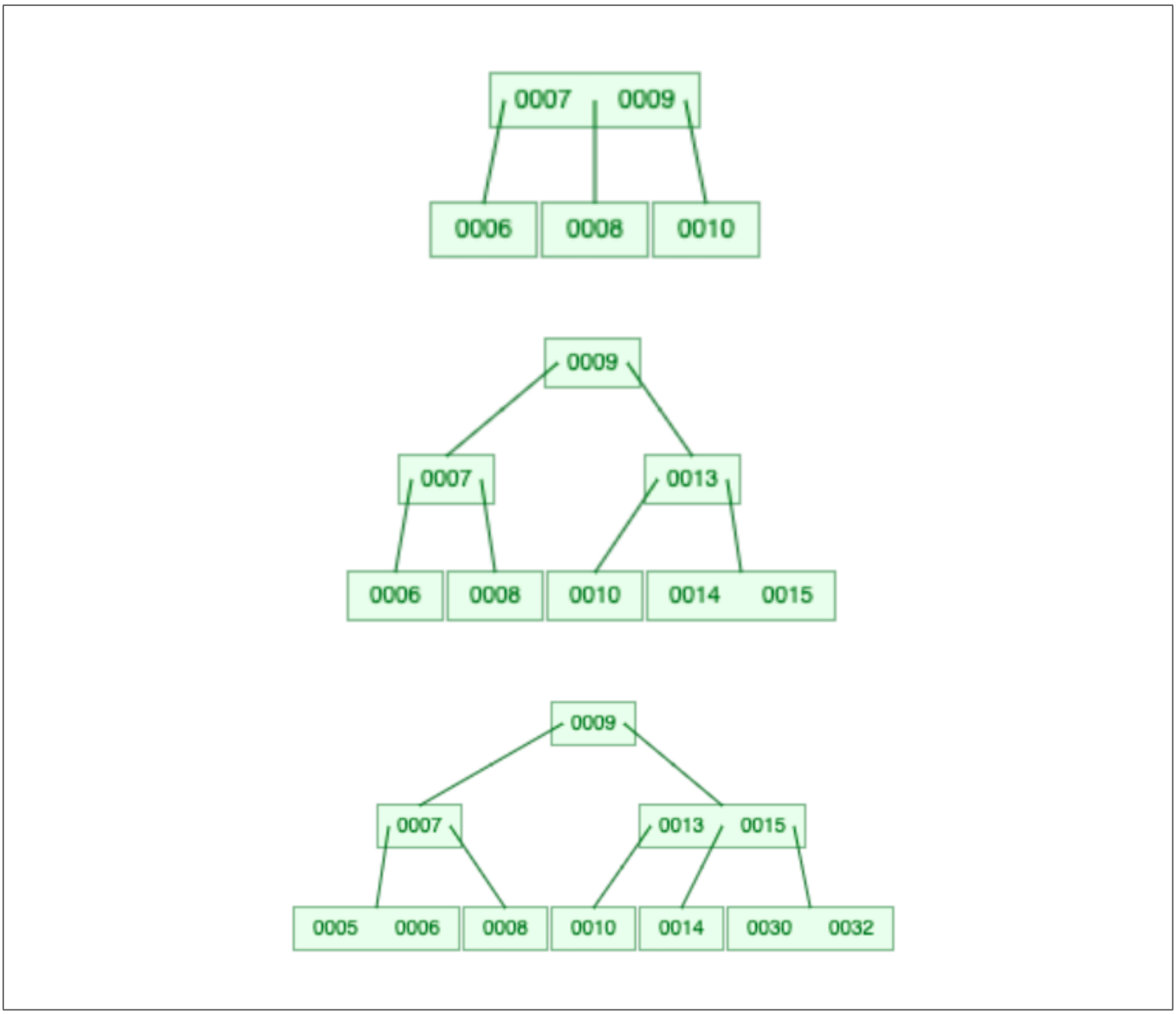
4. inserimento di $L/100 = 40.000$ ennuple, in ordine casuale, con valori di C pure uniformemente distribuiti (quindi si può supporre che vengano inserite 10 ennuple per ogni valore di C)
5. `SELECT * FROM R WHERE C = 100`

I 10 nuovi record con `C` $= 100$ si troveranno presumibilmente in 10 blocchi diversi e quindi il numero di accessi sarà pari al valore indicato nella risposta al punto precedente più 10 e quindi circa 23 o 24. L'indice, come nel caso precedente, si visita una volta sola

6. `CLUSTER` (questa operazione riordina la relazione; descrivere brevemente l'algoritmo utilizzato e il relativo costo; supporre una disponibilità di 1000 pagine di buffer)

La porzione disordinata è di circa 400 blocchi (numero di record inseriti dopo l'ordinamento $L/100 = 40.000$ diviso il fattore di blocco $B/r = 100$).
Si può pensare ad una variante del merge-sort, in cui vengono caricati nelle pagine del buffer tutti i blocchi con i record disordinati (lo spazio è sufficiente, perché sono 400, con 1000 pagine disponibili).
Questi record possono essere ordinati in memoria, con un costo che possiamo considerare trascurabile rispetto agli accessi in memoria secondaria. Poi si può procedere al merge, scorrendo la porzione già ordinata che può essere caricata via via, con un costo che è quindi pari al numero di blocchi.
Il tutto deve poi essere scritto.
Il costo totale è quindi pari a due volte il numero di blocchi del file, che deve essere letto per intero e poi riscritto:
$2 \times (L + L/100)/(B/r) = 80.800$

Domanda 3 (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: 10, 9, 8, 7, 6, 13, 14, 15, 30, 32, 5. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.



Domanda 4 (25%)
Si consideri una base di dati con le relazioni

- R1(A,B,C,D) con
  - vincolo di integrità referenziale fra l'attributo D e la chiave E della relazione R2
  - N_1=2.000.000 ennuple e fattore di blocco f_1=200
  - b=200 valori diversi sull'attributo B (tutti gli interi compresi fra 1 e b)
  - c=200.000 valori diversi sull'attributo C (tutti gli interi compresi fra 1 e c)
  - una struttura disordinata, un indice sulla chiave primaria A e un altro sull'attributo C;
- R2(E,F,G) con
  - N_2=1.000.000 ennuple e fattore di blocco f_2=10
  - una struttura disordinata, un indice sulla chiave primaria E

Supponendo che:

- gli indici abbiano tutti p=4 livelli (contando anche radice e foglie) e fattore di blocco massimo f_i=100
- il sistema esegua i join come **nested loop** (eventualmente utilizzando indici) oppure come **hash join**
- le pagine di buffer disponibili siano circa q=1000,

valutare il costo (indicandolo in modo sia simbolico sia numerico) di ciascuna delle interrogazioni seguenti **indicando anche il metodo di join che viene utilizzato**:

```
select *
from R1 join R2 on D=E
```

$$3 \times (\frac{N_1}{f_1} + \frac{N_2}{f_2}) = 330.000$$

- hash join

```
select *
from R1 join R2 on D=E
where B=5
```

$$\frac{N_1}{f_1} + \frac{N_1}{b}(p + 1 - 2) = 40.000$$

- scansione di R1 per la selezione: $\frac{N_1}{f_1}$
- un accesso diretto a R2 per ogni ennupla nel risultato della selezione

```
select *
from R1 join R2 on D=E
where C=502
```

$$p + \frac{N_1}{c} + \frac{N_1}{c}(p - 1 + 1) = \text{ ca. } 45$$

- accesso diretto a R1 per la selezione: p per l'indice e $\frac{N_1}{c}$ per le ennuple
- un accesso diretto a R2 per ogni ennupla nel risultato della selezione (radice resta nel buffer, ma nient'altro)