

# Esercizi

Si consideri la relazione  $R(\underline{A}, B, C, \dots)$  con la seguente configurazione

- $T(R) = 100.000.000$  tuple, ognuna di 40 byte
- A di 4 byte, con valori consecutivi da 1 a 100.000.000
- $B = 4000$ , il blocco B ha dimensione 4000 byte
- $P = 6$  byte, puntatore al blocco occupa 6 byte
- $P_r = 7$ , puntatore al record occupa 7 byte

Indicare la stima di costo per la seguente operazione:

1. `SELECT * FROM R WHERE A >= 1000 AND A <= 4000`

Considerando che file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

La stima di costo è la somma di 3 costi:

- (A) Costo B+Tree per cercare puntatore a A=1000
- (B) Costo lettura blocchi nodi foglia B+-tree (per A >= 1000 AND A <=4000)
- (C) costo lettura delle singole tuple nei file dati

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

(A) Costo B+Tree per cercare puntatore A=1000

**Definisco B+-tree sull'attributo A:**

Calcolo  $p_{\text{interno}}$  massimo su blocco di 4.000 byte:

Struttura  $p_{\text{interno}}$ : (p-1) chiavi + p puntatori al blocco

$$(p-1)*A + p*P \leq 4.000$$

$$(p-1)*4 + p*6 \leq 4.000$$

$$4p - 4 + 6p \leq 4.000$$

$$10p \leq 4004 \quad p = 400$$

Calcolo  $p_{\text{foglia}}$  massimo su blocco di 4.000 byte:

Struttura  $p_{\text{foglia}}$ : p chiavi + p puntatori al record + 1 puntatore blocco

$$p*A + pPr + P \leq 4.000$$

$$4p + 7p + 6 \leq 4.000$$

$$11p \leq 3994 \quad p = 363$$

Con riempimento 70%:  $p_{\text{interno}} = 400 * 0.70 = 280$

$p_{\text{foglia}} = 363 * 0.70 = 254$

Livello	#nodi	#valori di chiavi	#puntatori record
Root	1	(280-1)=279	-
1 liv.	280	(280x279) = 78.120	-
2 liv.	280 <sup>2</sup>	(280 <sup>2</sup> x279) = 21.873.600	-
3 liv./foglie	280 <sup>3</sup>	(280 <sup>3</sup> x254)=5.575.808.000	-

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

(A) Costo B+Tree per cercare puntatore A=1000: 4 operazioni I/O

**Definisco B<sup>+</sup>-tree sull'attributo A:**

Calcolo  $p_{\text{interno}}$  massimo su blocco di 4.000 byte:

Struttura  $p_{\text{interno}}$ : (p-1) chiavi + p puntatori al blocco

$$(p-1)*A + p*P \leq 4.000$$

$$(p-1)*4 + p*6 \leq 4.000$$

$$4p - 4 + 6p \leq 4.000$$

$$10p \leq 4004 \quad p = 400$$

Calcolo  $p_{\text{foglia}}$  massimo su blocco di 4.000 byte:

Struttura  $p_{\text{foglia}}$ : p chiavi + p puntatori al record + 1 puntatore blocco

$$p*A + pPr + P \leq 4.000$$

$$4p + 7p + 6 \leq 4.000$$

$$11p \leq 3994 \quad p = 363$$

Con riempimento 70%:  $p_{\text{interno}} = 400 * 0.70 = 280$

$p_{\text{foglia}} = 363 * 0.70 = 254$

Livello	#nodi	#valori di chiavi	#puntatori record
Root	1	(280-1)=279	-
1 liv.	280	(280x279) = 78.120	-
2 liv.	280 <sup>2</sup>	(280 <sup>2</sup> x279) = 21.873.600	-
3 liv./foglie	280 <sup>3</sup>	(280 <sup>3</sup> x254)=5.575.808.000	-

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

(B) Costo lettura blocchi nodi foglia B+-tree (per A >= 1000 AND A <=4000)

Devo leggere i puntatori ai record di circa 3000 tuple

Ogni nodo foglia contiene 254 valori di chiave con I corrispondenti puntatori,

I valori di chiave di 3000 tuple richiedono

$3000/254 = 12$  nodi foglia

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

(B) Costo lettura blocchi nodi foglia B+-tree (per A >= 1000 AND A <=4000): 12 operazioni I/O

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

(C) Costo lettura delle singole tuple nei file dati

Devo leggere 3001 tuple su file non ordinato

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

(C) Costo lettura delle singole tuple nei file dati: 3001 operazioni I/O

Devo leggere 3001 tuple su file non ordinato



# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file non è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

La stima di costo è la somma di 3 costi:

3017 I/O

(A) Costo B+Tree per cercare puntatore a A=1000:

4 I/O

(B) Costo lettura blocchi nodi foglia B+-tree (per A >= 1000 AND A <=4000):

12 I/O

(C) costo lettura delle singole tuple nei file dati:

3001 I/O

# Esercizi

Si consideri la relazione  $R(\underline{A}, B, C, \dots)$  con la seguente configurazione

- $T(R) = 100.000.000$  tuple, ognuna di 40 byte
- A di 4 byte, con valori consecutivi da 1 a 100.000.000
- $B = 4000$ , il blocco B ha dimensione 4000 byte
- $P = 6$  byte, puntatore al blocco occupa 6 byte
- $P_r = 7$ , puntatore al record occupa 7 byte

Indicare la stima di costo per la seguente operazione:

1. `SELECT * FROM R WHERE A >= 1000 AND A <= 4000`

Considerando che file è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

# Esercizi

SELECT \* FROM R WHERE A  $\geq$  1000 AND A  $\leq$  4000

file è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

La stima di costo è la somma di 3 costi:

- (A) Costo B+Tree per cercare puntatore a A=1000:
- (B) Costo lettura blocchi nodi foglia B+-tree (per A  $\geq$  1000 AND A  $\leq$  4000):
- (C) costo lettura delle singole tuple nei file dati:

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

La stima di costo è la somma di 3 costi:

- (A) Costo B+Tree per cercare puntatore a A=1000: 4 I/O
- (B) Costo lettura blocchi nodi foglia B+-tree (per A >= 1000 AND A <=4000): 12 I/O
- (C) costo lettura delle singole tuple nei file dati:

Devo leggere 3001 tuple su file ordinato. Le tuple sono memorizzate contigue, quindi

Fattore blocco:  $4000/40=100$

3001 tuple contigue richiedono di leggere  $3001/100= 31$  blocchi

# Esercizi

SELECT \* FROM R WHERE A >= 1000 AND A <=4000

file è ordinato su A, con indice B+-tree su A con fattore di riempimento al 70%

La stima di costo è la somma di 3 costi:

47 I/O

(A) Costo B+Tree per cercare puntatore a A=1000:

4 I/O

(B) Costo lettura blocchi nodi foglia B+-tree (per A >= 1000 AND A <=4000):

12 I/O

(C) costo lettura delle singole tuple nei file dati:

31 I/O

Devo leggere 3001 tuple su file ordinato. Le tuple sono memorizzate contigue, quindi

Fattore blocco:  $4000/40=100$

3001 tuple contigue richiedono di leggere  $3001/100= 31$  blocchi

# Esercizi

Si consideri due relazioni R e S con la seguente configurazione

- R e S non hanno indici
- R occupa 9 blocchi identificati con i seguenti indirizzi:  $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9$
- S occupa 4 blocchi identificati con i seguenti indirizzi:  $S_1, S_2, S_3, S_4$
- Il buffer ha a disposizione 6 blocchi in tutto per l'esecuzione del join

Considerare l'esecuzione ottimizzata di un join tra R e S con nested loop.

Rispondere alla seguente domanda:

- Indicare nell'ordine di esecuzione le operazioni di carico dei blocchi nel buffer (i.e., fix), specificando l'indirizzo del blocco (e.g., fix( $R_8$ )) e le operazioni di scarico (e.g., unfix( $R_8$ ))

# Esercizi

Si consideri due relazioni R e S con la seguente configurazione

- R e S non hanno indici
- R occupa 9 blocchi identificati con i seguenti indirizzi:  $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9$
- S occupa 4 blocchi identificati con i seguenti indirizzi:  $S_1, S_2, S_3, S_4$
- Il buffer ha a disposizione 6 blocchi in tutto per l'esecuzione del join

Considerare l'esecuzione ottimizzata di un join tra R e S con nested loop. Rispondere alla seguente domanda:

- Indicare nell'ordine di esecuzione le operazioni di carico dei blocchi nel buffer (i.e., fix), specificando l'indirizzo del blocco (e.g., fix( $R_8$ )) e le operazioni di scarico (e.g., unfix( $R_8$ ))

- Si esegue S  $\bowtie$  R con nested loop join

fix( $S_1$ ), fix( $S_2$ ), fix( $S_3$ ), fix( $S_4$ ), fix( $R_1$ ),

unfix( $R_1$ ), fix( $R_2$ ),

unfix( $R_2$ ), fix( $R_3$ ),

unfix( $R_3$ ), fix( $R_4$ ),

unfix( $R_4$ ), fix( $R_5$ ),

unfix( $R_5$ ), fix( $R_6$ ),

unfix( $R_6$ ), fix( $R_7$ ),

unfix( $R_7$ ), fix( $R_8$ ),

unfix( $R_8$ ), fix( $R_9$ ), unfix( $R_9$ ),

unfix( $S_1$ ), unfix( $S_2$ ), unfix( $S_3$ ), unfix( $S_4$ ), unfix( $R_1$ )

# Esercizi

Si consideri due relazioni R e S con la seguente configurazione

- R e S non hanno indici
- R occupa 9 blocchi identificati con i seguenti indirizzi:  $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9$
- S occupa 4 blocchi identificati con i seguenti indirizzi:  $S_1, S_2, S_3, S_4$
- Il buffer ha a disposizione **5 blocchi** in tutto per l'esecuzione del join

Considerare l'esecuzione ottimizzata di un join tra R e S con nested loop.

Rispondere alla seguente domanda:

- Indicare nell'ordine di esecuzione le operazioni di carico dei blocchi nel buffer (i.e., fix), specificando l'indirizzo del blocco (e.g.,  $\text{fix}(R_8)$ ) e le operazioni di scarico (e.g.,  $\text{unfix}(R_8)$ )



# Esercizi

Si consideri due relazioni R e S con la seguente configurazione

- R e S non hanno indici
- R occupa 9 blocchi identificati con i seguenti indirizzi:  $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9$
- S occupa 4 blocchi identificati con i seguenti indirizzi:  $S_1, S_2, S_3, S_4$
- Il buffer ha a disposizione **5 blocchi** in tutto per l'esecuzione del join

Considerare l'esecuzione ottimizzata di un join tra R e S con nested loop. Rispondere alla seguente domanda:

- Indicare nell'ordine di esecuzione le operazioni di carico dei blocchi nel buffer (i.e., fix), specificando l'indirizzo del blocco (e.g.,  $\text{fix}(R_8)$ ) e le operazioni di scarico (e.g.,  $\text{unfix}(R_8)$ )

- Si esegue  $S \bowtie R$  con nested loop join

$\text{fix}(S_1), \text{fix}(S_2), \text{fix}(S_3), \text{fix}(R_1),$

$\text{unfix}(R_1), \text{fix}(R_2),$

$\text{unfix}(R_2), \text{fix}(R_3),$

$\text{unfix}(R_3), \text{fix}(R_4),$

$\text{unfix}(R_4), \text{fix}(R_5),$

$\text{unfix}(R_5), \text{fix}(R_6),$

$\text{unfix}(R_6), \text{fix}(R_7),$

$\text{unfix}(R_7), \text{fix}(R_8),$

$\text{unfix}(R_8), \text{fix}(R_9),$

$\text{unfix}(S_1), \text{unfix}(S_2),$

$\text{unfix}(S_3), \text{fix}(S_4),$

$\text{unfix}(R_9), \text{fix}(R_1),$

$\text{unfix}(R_1), \text{fix}(R_2),$

$\text{unfix}(R_2), \text{fix}(R_3),$

$\text{unfix}(R_3), \text{fix}(R_4),$

$\text{unfix}(R_4), \text{fix}(R_5),$

$\text{unfix}(R_5), \text{fix}(R_6),$

$\text{unfix}(R_6), \text{fix}(R_7),$

$\text{unfix}(R_7), \text{fix}(R_8),$

$\text{unfix}(R_8), \text{fix}(R_9),$

$\text{unfix}(R_9), \text{unfix}(S_4)$

# Esercizi

Si consideri due relazioni R e S con la seguente configurazione

- $T(R)=900$
- $T(S)=400$
- R occupa 9 blocchi identificati con i seguenti indirizzi:  $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9$
- S occupa 4 blocchi identificati con i seguenti indirizzi:  $S_1, S_2, S_3, S_4$
- Condizione di join  $R.A=S.B$ , su A è costruito un indice A i cui blocchi risiedono tutti nel buffer
- Il buffer ha a disposizione 5 blocchi in tutto per l'esecuzione del join

Considerare l'esecuzione ottimizzata di un join tra R e S con nested loop con indice. Rispondere alla seguente domanda:

- Indicare nell'ordine di esecuzione le operazioni di carico dei blocchi nel buffer (i.e., fix), specificando l'indirizzo del blocco (e.g.,  $\text{fix}(R_8)$ ) e le operazioni di scarico (e.g.,  $\text{unfix}(R_8)$ )
- **Si esegue S ⋈ R con nested loop join e indice su R**

Per ogni tupla  $s'$  in  $T(S)$  attraverso l'indice si cerca il blocco di  $R_j$ , dove è memorizzata la tupla  $r'$ ,  $tc\ r'.A=s'.B$  e si legge il blocco  $R_j$

- dato che i blocchi dell'indice sono già nel buffer, non ci sono operazioni I/O per l'indice, c'è solo la lettura di  $R_j$

$\text{fix}(S_1), \text{fix}(S_2), \text{fix}(S_3), \text{fix}(S_4) + 400 \text{ volte } \text{fix}(R_j), \text{unfix}(R_j)$