

**COMPITO DI SISTEMI INFORMATIVI/BASI DI DATI II**

**8 marzo 2007 (Tot. 16) Tempo: 2h**

**Esercizio 1 (punti 3)**

Si consideri il seguente log:

- |                      |                      |
|----------------------|----------------------|
| 1. I(T2,O1,A1)       | 16. I(T4,O6,A16)     |
| 2. B(T1)             | 17. CK(T1,T3,T4,T5)  |
| 3. U(T1,O1,B3,A3)    | 18. C(T1)            |
| 4. D(T2,O1,B4)       | 19. B(T6)            |
| 5. B(T3)             | 20. U(T3,O4,B20,A20) |
| 6. I(T3,O2,A6)       | 21. D(T6,O4,B21)     |
| 7. U(T2,O2,B7,A7)    | 22. B(T7)            |
| 8. C(T2)             | 23. I(T7,O7,A23)     |
| 9. D(T1,O3,B9)       | 24. C(T7)            |
| 10. U(T3,O4,B10,A10) | 25. D(T3,O5,B25)     |
| 11. B(T4)            | 26. U(T5,O7,B26,A26) |
| 12. U(T4,O4,B12,A12) | 27. I(T6,O8,A27)     |
| 13. I(T1,O5,A13)     | 28. U(T3,O6,B28,A28) |
| 14. B(T5)            | 29. D(T5,O6,B29)     |
| 15. U(T5,O5,B15,A15) | 30. C(T5)            |

si mostrino le operazioni di recovery da effettuare supponendo che il guasto avvenga subito dopo l'ultimo record del log.

**Esercizio 2 (punti 4)**

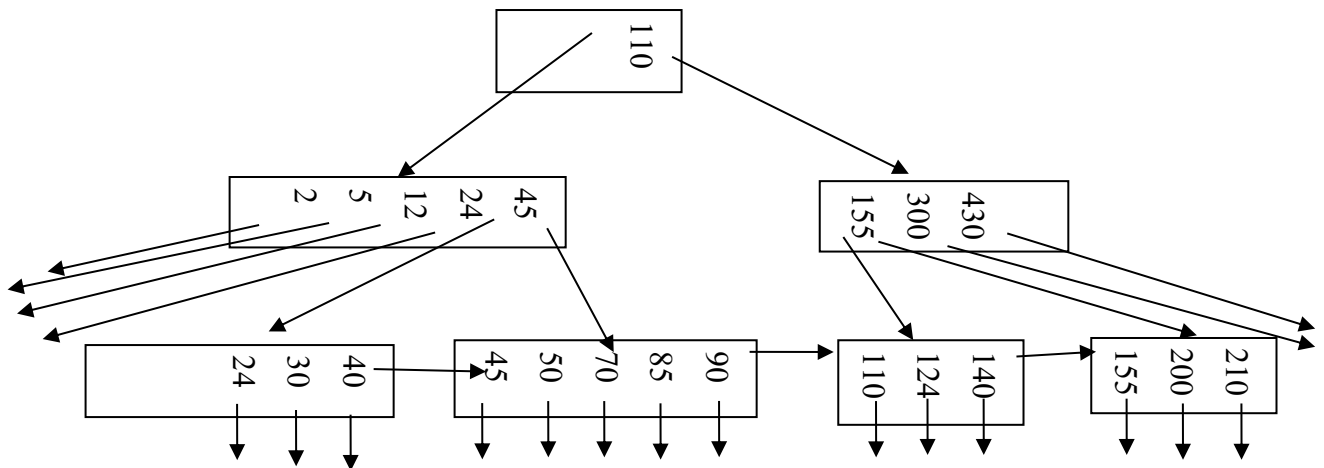
Dato il seguente schedule:

r2(x) w3(y) w1(x) r5(p) w5(x) w2(z) r4(w) r3(z) r5(y) w4(p) w2(w)

si indichi se e' conflict serializzabile.

### Esercizio 3 (punti 3)

Sia dato il seguente B+ tree di ordine 5.



Si mostri come si modifica l'albero nel caso di inserimento della chiave 60

### Esercizio 4 (punti 6)

Siano date le relazioni

Vendita(CodVendita,Data,Negozio, CodArticolo, Quantità)

Articolo(CodArticolo,Descrizione,Prezzo)

la query

```
SELECT V.*, A.*
```

```
FROM Vendita AS V, Articolo AS A
```

```
WHERE V.CodArticolo=A.CodArticolo AND A.Prezzo>10 AND A.Prezzo<15 AND  
V.Data='1/1/2000'
```

e i parametri:

buffer di memoria centrale disponibili:  $M=10.000$

dimensione del buffer:  $B=4.000$  bytes

numero di tuple:  $T(\text{Vendita})= 1.000.000$ ,  $T(\text{Articolo})= 480.000$ ,

dimensione delle tuple:  $S(\text{Vendita})=800$  bytes,  $S(\text{Articolo})=500$ ,

$S(\text{CodArticolo})=100$

numero di valori:  $V(\text{Vendita},\text{Data})=10$   $V(\text{Vendita},\text{CodArticolo})=8.000$

Si calcoli il costo minimo di ciascuna delle seguenti sequenze:

$(\sigma_{\text{Prezzo}>10 \text{ AND } \text{Prezzo}<15} \text{Articolo}) \triangleright \triangleleft (\sigma_{\text{Data}='1/1/2000'} \text{Vendita})$

$\sigma_{\text{Data}='1/1/2000 \text{ AND } \text{Prezzo}>10 \text{ AND } \text{Prezzo}<15}(\text{Articolo}) \triangleright \triangleleft \text{Vendita}$

Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record. Si supponga di avere un indice primario su Articolo.Prezzo e un indice secondario su Vendita.Data. CodArticolo è chiave primario di Articolo. Per il calcolo del costo e della cardinalità della selezione sul Prezzo si utilizzi la stima dei valori nel range, sapendo che il range di Prezzo è  $[9,20]$ .

## SOLUZIONE

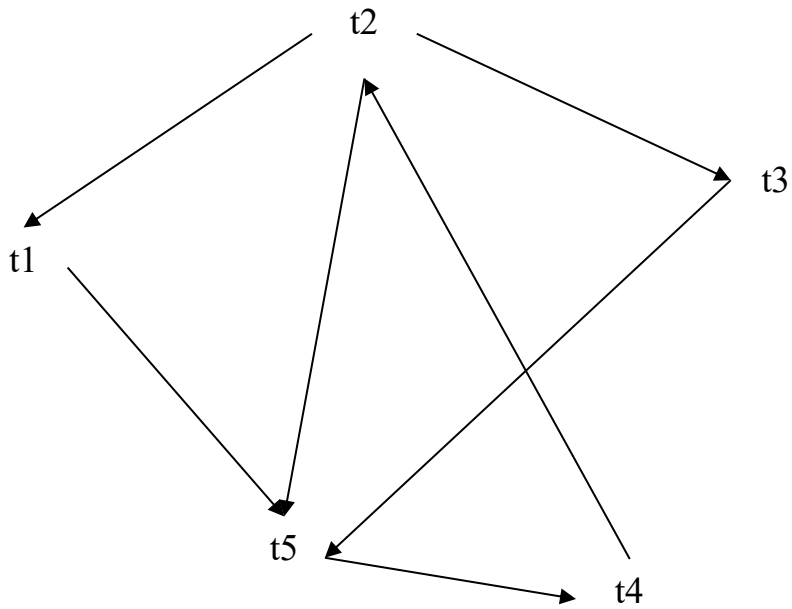
### Esercizio 1

1. I(T2,O1,A1)
2. B(T1)
3. U(T1,O1,B3,A3)
4. D(T2,O1,B4)
5. B(T3)
6. I(T3,O2,A6)
7. U(T2,O2,B7,A7)
8. C(T2)
9. D(T1,O3,B9)
10. U(T3,O4,B10,A10)
11. B(T4)
12. U(T4,O4,B12,A12)
13. I(T1,O5,A13)
14. B(T5)
15. U(T5,O5,B15,A15)
16. I(T4,O6,A16)
17. CK(T1,T3,T4,T5)
18. C(T1)
19. B(T6)
20. U(T3,O4,B20,A20)
21. D(T6,O4,B21)
22. B(T7)
23. I(T7,O7,A23)
24. C(T7)
25. D(T3,O5,B25)
26. U(T5,O7,B26,A26)
27. I(T6,O8,A27)
28. U(T3,O6,B28,A28)
29. D(T5,O6,B29)
30. C(T5)

- 17 UNDO={T1,T3,T4,T5} REDO={}
- 18 UNDO={T3,T4,T5} REDO={T1}
- 19 UNDO={T3,T4,T5,T6} REDO={T1}
- 22 UNDO={T3,T4,T5,T6,T7} REDO={T1}
- 24 UNDO={T3,T4,T5,T6} REDO={T1,T7}
- 30 UNDO={T3,T4, T6} REDO={T1,T5,T7}
- UNDO
- 28 O6=B28
- 27 D(O8)
- 25 I(O5,B25)
- 21 I(O4,B21)
- 20 O4=B20
- 16 D(O6)
- 12 O4=B12
- 10 O4=B10
- 6 D(O2)
- REDO
- 3 O1=A3
- 9 D(O3)
- 13 I(O5,A13)
- 15 O5=A15
- 23 I(O7,A23)
- 26 O7=A26
- 29 D(O6)

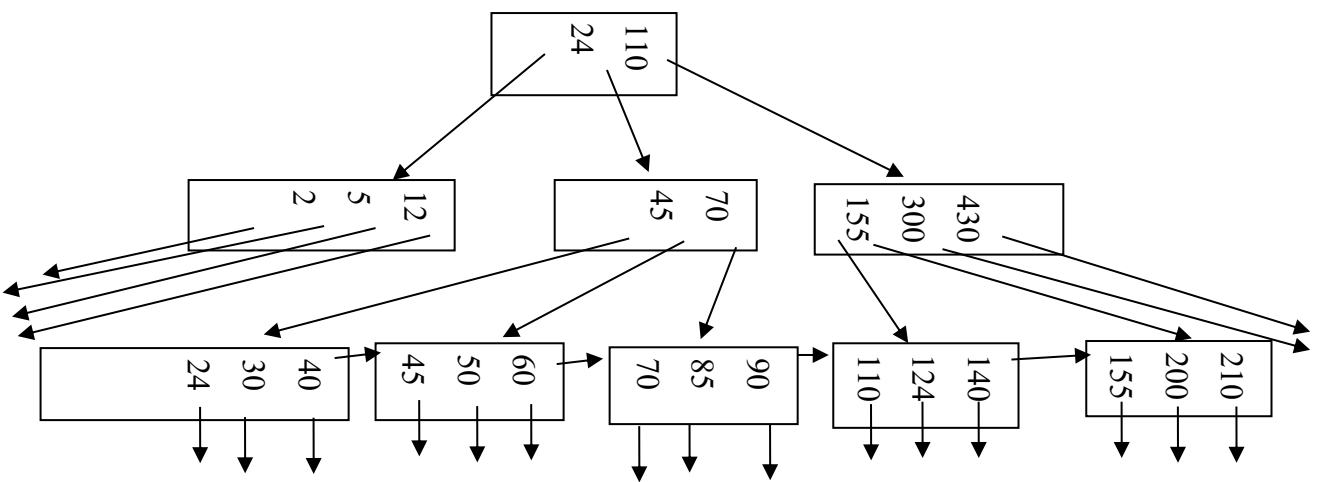
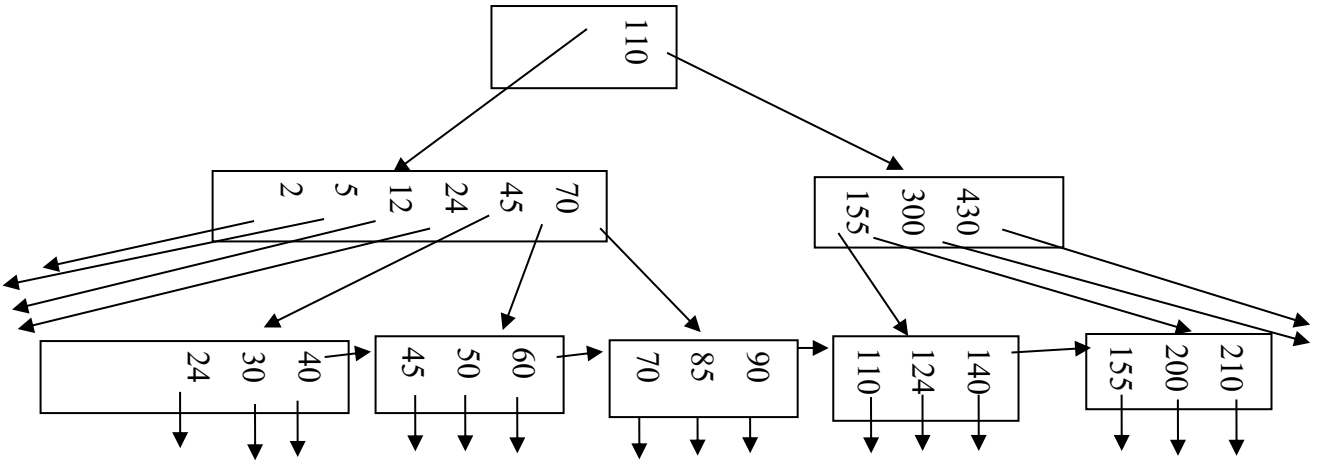
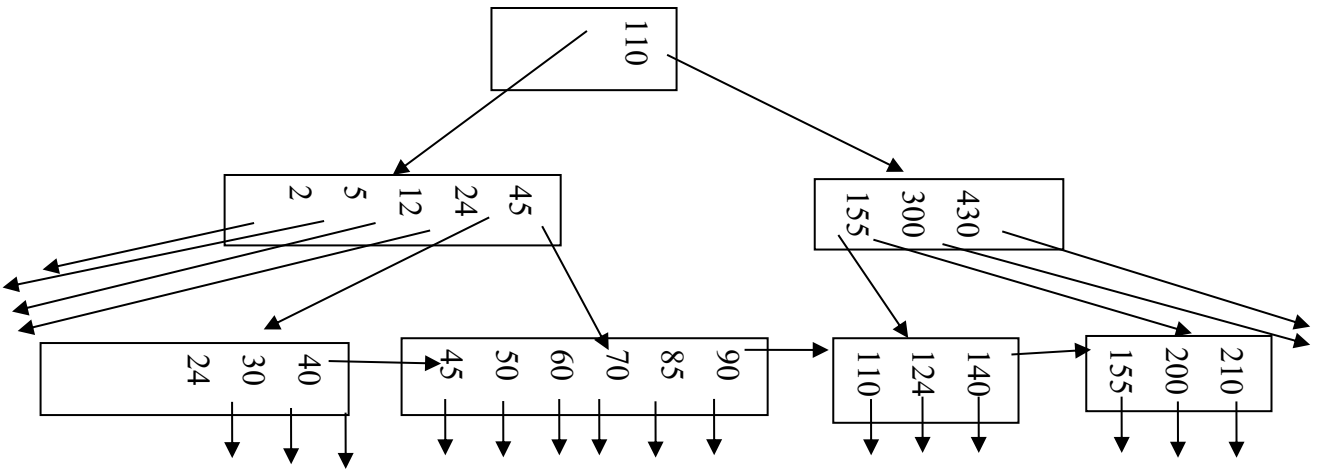
**Esercizio 2**

r2(x) w3(y) w1(x) r5(p) w5(x) w2(z) r4(w) r3(z) r5(y) w4(p) w2(w)



Lo schedule non è conflict serializzabile perché il grafo presenta un ciclo.

### Esercizio 3



#### Esercizio 4

Calcoliamo innanzitutto il numero di blocchi occupati da ciascuna relazione:

$$B(\text{Vendita}) = T(\text{Vendita}) * S(\text{Vendita}) / B = 1000000 * 800 / 4000 = 200000$$

$$B(\text{Articolo}) = 480000 * 500 / 4000 = 60000$$

Prima sequenza:  $(\sigma_{\text{Prezzo} > 10 \text{ AND } \text{Prezzo} < 15} \text{Articolo}) \bowtie (\sigma_{\text{Data} = '1/1/2000'} \text{Vendita})$

Sia  $X = \sigma_{\text{Prezzo} > 10 \text{ AND } \text{Prezzo} < 15} \text{Articolo}$

Utilizzando l'indice primario su Prezzo

$$f = (15 - 10 - 1) / (20 - 9 + 1) = 4 / 12 = 0.333$$

$$\text{CostoIndice}(X) = 3 + \lceil f * B(\text{Articolo}) \rceil = 3 + 0.333 * 60000 = 20003$$

$$\text{CostoScansione}(X) = B(\text{Articolo}) = 60000$$

Il costo dell'accesso con indice è inferiore

$$T(X) = f * T(\text{Articolo}) = 0.333 * 480000 = 159840$$

$$B(X) = 159840 * 500 / 4000 = 19980$$

Sia  $Y = \sigma_{\text{Data} = '1/1/2000'} \text{Vendita}$

Utilizzando l'indice secondario su Data

$$\text{Costo}(Y) = 3 + T(\text{Vendita}) / V(\text{Vendita}, \text{Data}) = 3 + 1000000 / 10 = 3 + 100000 = 100003$$

$$T(Y) = T(\text{Vendita}) / V(\text{Vendita}, \text{Data}) = 1000000 / 10 = 100000$$

$$B(Y) = 100000 * 800 / 4000 = 20000$$

Né X né Y stanno interamente in memoria centrale quindi provo con il join nested-loop basato sui blocchi e con l'hash join ibrido

$$\text{CostoNLJ}(X \bowtie Y) = B(Y) + B(X) * B(Y) / M = 19980 + 19980 * 20000 / 10000 = 59940$$

$$\text{CostoHJ}(X \bowtie Y) = (3 - 2M / B(Y)) * (B(X) + B(Y)) = (3 - 20000 / 19980) * (30000 + 19980) = 99910$$

$$\text{Costo}((\sigma_{\text{Prezzo} > 10 \text{ AND } \text{Prezzo} < 15} \text{Articolo}) \bowtie (\sigma_{\text{Data} = '1/1/2000'} \text{Vendita})) = 20003 + 100003 + 59940 = 179946$$

Seconda sequenza:  $\sigma_{\text{Data} = '1/1/2000' \text{ AND } \text{Prezzo} > 10 \text{ AND } \text{Prezzo} < 15} (\text{Articolo} \bowtie \text{Vendita})$

Sia  $Z = \text{Articolo} \bowtie \text{Vendita}$

Né Articolo né Vendita stanno in memoria centrale, quindi provo con il join nested-loop basato sui blocchi e con l'hash join ibrido

Join nested-loop basato sui blocchi:

$$\text{CostoNLJ}(Z) = B(\text{Articolo}) + B(\text{Articolo}) * B(\text{Vendita}) / M = 60000 + 60000 * 200000 / 10000 = 1260000$$

Hash join ibrido: si può fare perchè  $10000 > \sqrt{60000} = 245$

$$\text{CostoHJ}(Z) = (3 - 2 * 10000 / 60000) * (60000 + 200000) = 693333$$

Il migliore è l'hash-join ibrido

$$T(Z) = T(\text{Articolo}) * T(\text{Vendita}) / \max\{V(\text{Articolo}, \text{CodArticolo}), V(\text{Vendita}, \text{CodArticolo})\} = 480000 * 1000000 / \max\{480000, 8000\} = 1000000$$

$$S(Z) = S(\text{Vendita}) + S(\text{Articolo}) - S(\text{CodArticolo}) = 800 + 500 - 100 = 1200$$

$$B(Z) = 1000000 * 1200 / 4000 = 300000$$

Dato che non ci sono indici su Z:

$$\text{Costo}(\sigma_{\text{Data} = '1/1/2000' \text{ AND } \text{Prezzo} > 10 \text{ AND } \text{Prezzo} < 15} Z) = B(Z) = 300000$$

Costo( $\sigma_{\text{Data}='1/1/2000 \text{ AND Prezzo}=10 \text{ AND Prezzo} < 15$  (Articolo  $\triangleright \triangleleft$  Vendita)) = 693333+300000=993333