

COMPITO DI SISTEMI INFORMATIVI

8 gennaio 2010 (Tot. 16) Tempo: 2h

Esercizio 1 (punti 3)

Si consideri il seguente log:

- | | |
|----------------------|----------------------|
| 1. U(T1,O1,B1,A1) | 15. I(T4,O4,A15) |
| 2. B(T2) | 16. U(T5,O4,B16,A16) |
| 3. U(T2,O1,B3,A3) | 17. I(T3,O5,A17) |
| 4. I(T1,O2,A4) | 18. B(T6) |
| 5. C(T1) | 19. U(T6,O5,B19,A19) |
| 6. B(T3) | 20. D(T5,O2,B20) |
| 7. U(T3,O2,B7,A7) | 21. C(T3) |
| 8. I(T2,O3,A8) | 22. U(T4,O3,B22,A22) |
| 9. D(T3,O1,B9) | 23. I(T6,O6,A23) |
| 10. B(T4) | 24. B(T7) |
| 11. U(T4,O3,B11,A11) | 25. C(T4) |
| 12. B(T5) | 26. C(T6) |
| 13. U(T5,O2,B13,A13) | 27. U(T7,O6,B27,A27) |
| 14. CK(T2,T3,T4,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)

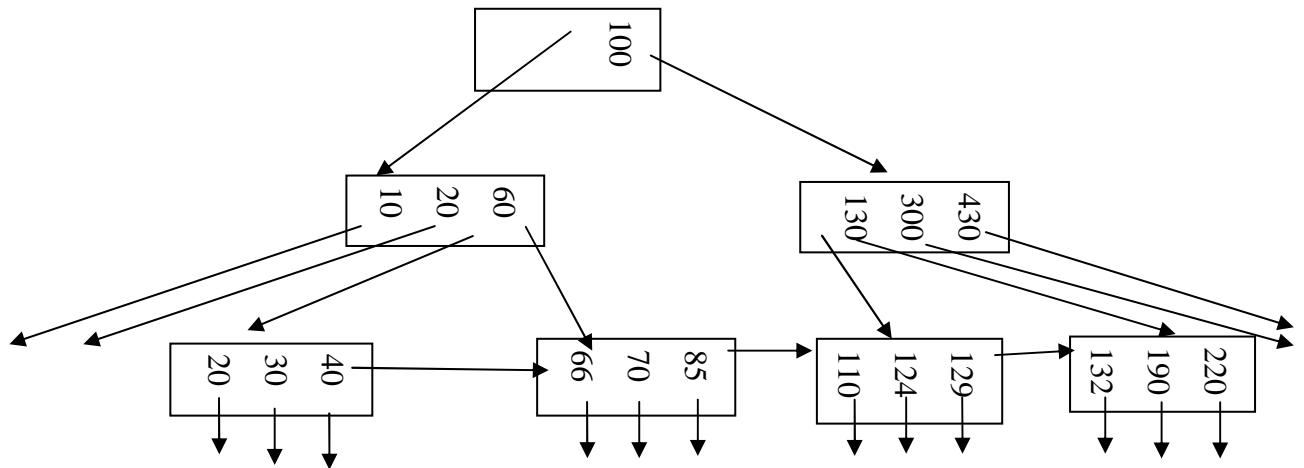
Dato il seguente schedule:

$r_1(x) r_2(y) w_2(x) r_2(w) w_1(z) w_3(w) w_4(y) r_3(z) r_2(z) r_4(w)$

si indichi se è view serializzabile e, se sì, si indichi uno schedule seriale view-equivalente.

Esercizio 3 (punti 3)

Sia dato il seguente B+ tree di ordine 3.



Si mostri come si modifica l'albero nel caso di inserimento delle chiavi 35 e 130

Esercizio 4 (punti 6)

Siano date le relazioni

Utente(CodiceUt, Nome, Cognome, Indirizzo, DataDiNascita, Fornitore)

Servizio(CodiceServ, NomeServ, Descrizione, Fornitore, Costo)

la query

```
SELECT U.*, S.*
```

```
FROM Utente AS U, Servizio AS S
```

```
WHERE U.Fornitore =S.Fornitore AND S.Costo>70
```

e i parametri:

buffer di memoria centrale disponibili: $M=50000$

dimensione del buffer: $B=4000$ bytes

numero di tuple: $T(\text{Utente})= 800000$, $T(\text{Servizio})= 200000$,

dimensione delle tuple: $S(\text{Utente})=500$ bytes, $S(\text{Servizio})=1600$,

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

valori: $V(\text{Utente}, \text{Fornitore})=4000$, $V(\text{Servizio}, \text{Fornitore})=20000$

range di Servizio.Costo= $[1,100]$

Si calcoli il costo minimo di ciascuna delle seguenti sequenze:

$\sigma_{\text{Costo}>70}(\text{Servizio} \bowtie \text{Utente})$

$(\sigma_{\text{Costo}>70} \text{ Servizio}) \bowtie \text{Utente}$

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 secondario su Servizio.Costo e che Utente e Servizio siano entrambe ordinate sul campo Fornitore.

SOLUZIONE

Esercizio 1

1. U(T1,O1,B1,A1)
2. B(T2)
3. U(T2,O1,B3,A3)
4. I(T1,O2,A4)
5. C(T1)
6. B(T3)
7. U(T3,O2,B7,A7)
8. I(T2,O3,A8)
9. D(T3,O1,B9)
10. B(T4)
11. U(T4,O3,B11,A11)
12. B(T5)
13. U(T5,O2,B13,A13)
14. CK(T2,T3,T4,T5)
15. I(T4,O4,A15)
16. U(T5,O4,B16,A16)
17. I(T3,O5,A17)
18. B(T6)
19. U(T6,O5,B19,A19)
20. D(T5,O2,B20)
21. C(T3)
22. U(T4,O3,B22,A22)
23. I(T6,O6,A23)
24. B(T7)
25. C(T4)
26. C(T6)
27. U(T7,O6,B27,A27)

- 14 UNDO={T2,T3,T4,T5} REDO={}
18 UNDO={T2,T3,T4,T5,T6} REDO={}
21 UNDO={T2,T4,T5,T6} REDO={T3}
24 UNDO={T2,T4,T5,T6,T7} REDO={T3}
25 UNDO={T2,T5,T6,T7} REDO={T3,T4}
26 UNDO={T2,T5,T7} REDO={T3,T4,T6}

UNDO

- 27 O6=B27
20 I(O2,B20)
16 O4=B16
13 O2=B13
8 D(O3)
3 O1=B3

REDO

- 7 O2=A7
9 D(O1)
11 O3=A11
15 I(O4,A15)
17 I(O5,A17)
19 O5=A19
22 O3=A22
23 I(O6,A23)

Esercizio 2

r1(x) r2(y) w2(x) r2(w) w1(z) w3(w) w4(y) r3(z) r2(z) r4(w)

si indichi se è view serializzabile e, se sì, si indichi uno schedule seriale view-equivalente.

"legge-da"={ (r3(z),w1(z)),(r2(z),w1(z)),(r4(w),w3(w)) }

scritture finali={ (t2,x),(t4,y),(t1,z),(t3,w) }

Schedule seriale

r1(x) w1(z) r2(y) w2(x) r2(w) r2(z) w3(w) r3(z) w4(y) r4(w)

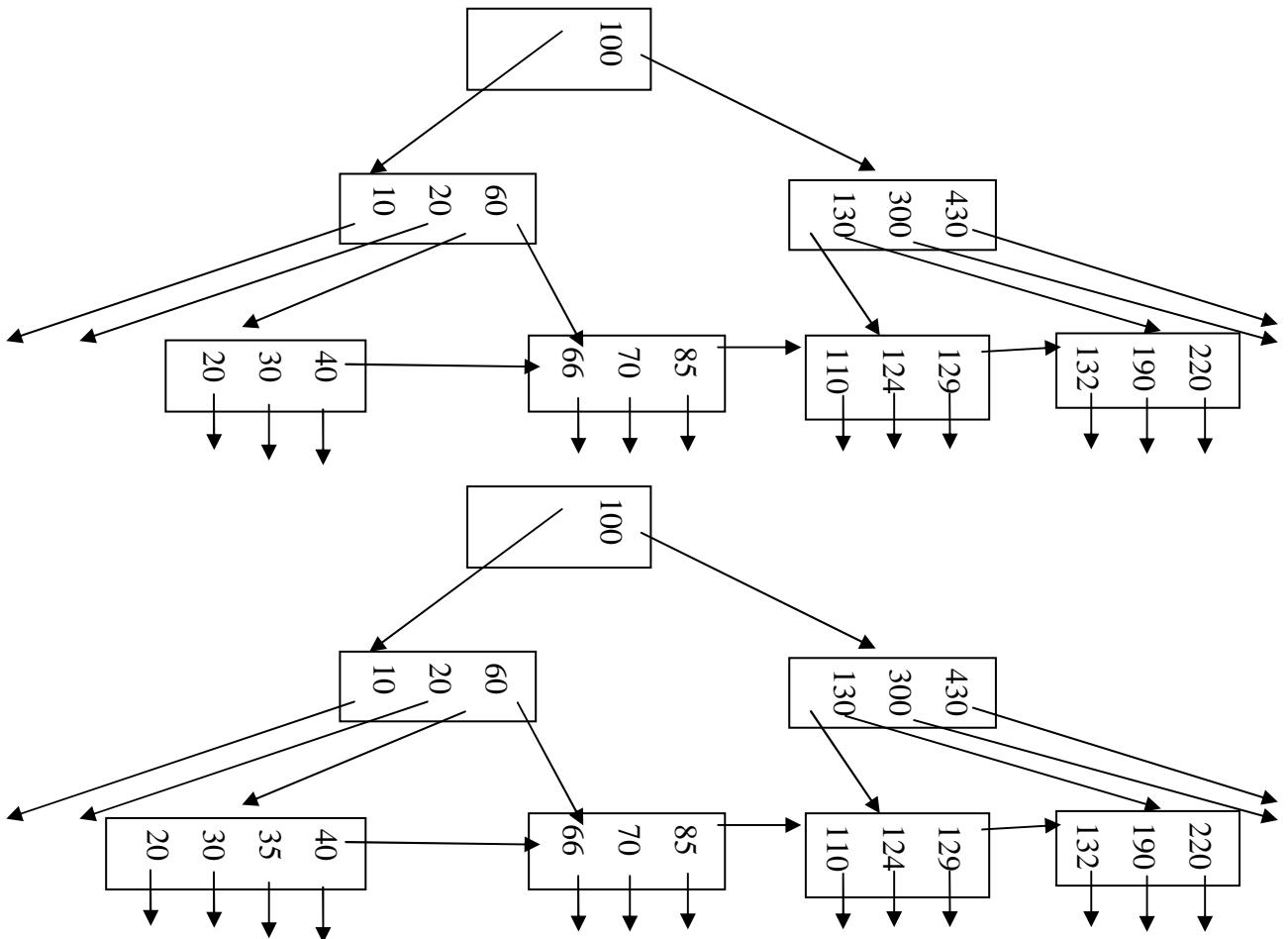
"legge-da"={ (r3(z),w1(z)),(r2(z),w1(z)),(r4(w),w3(w)) }

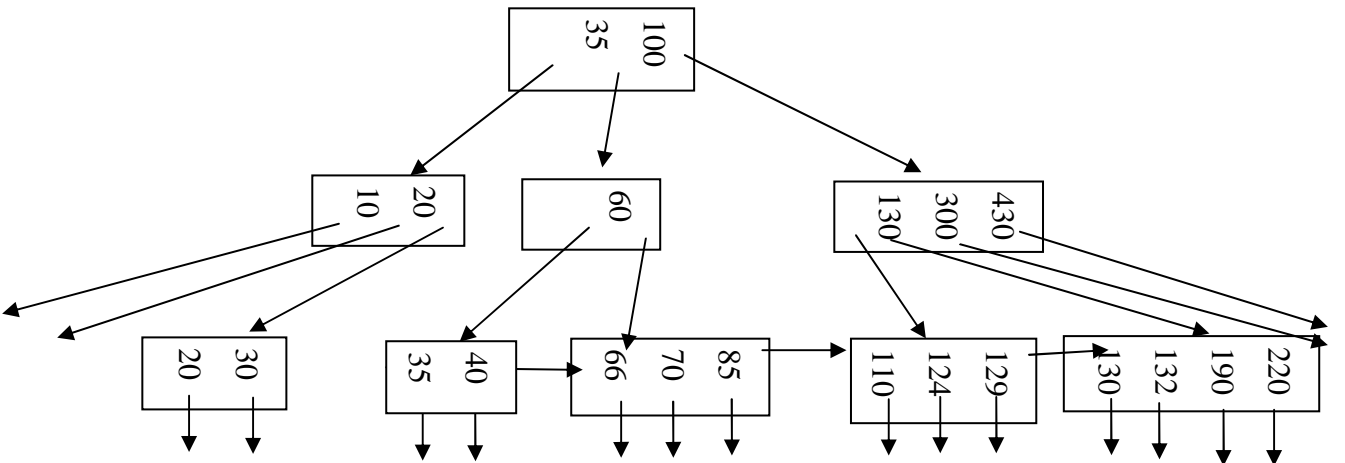
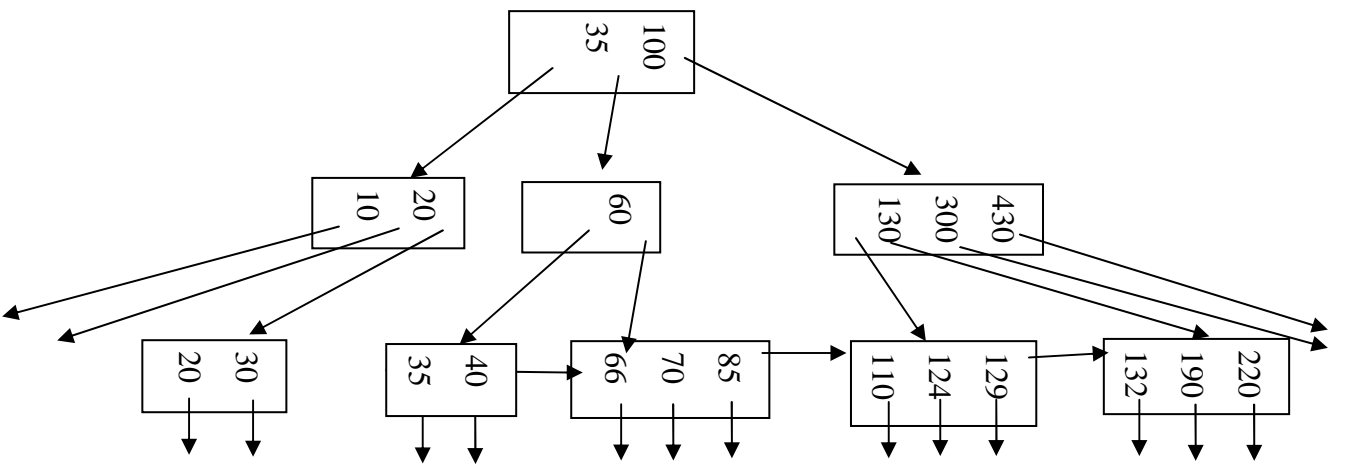
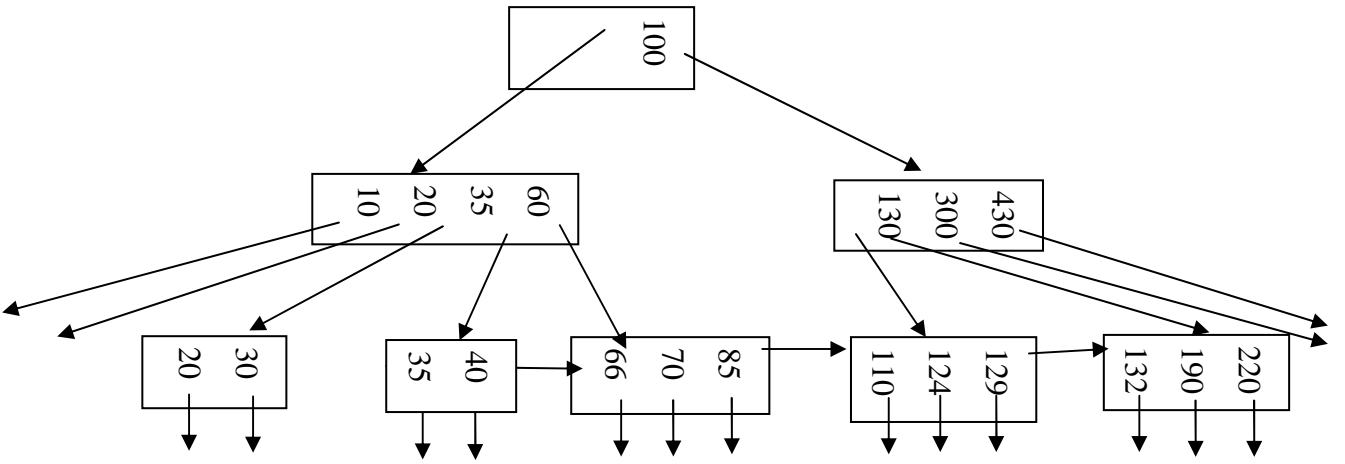
scritture finali={ (t2,x),(t4,y),(t1,z),(t3,w) }

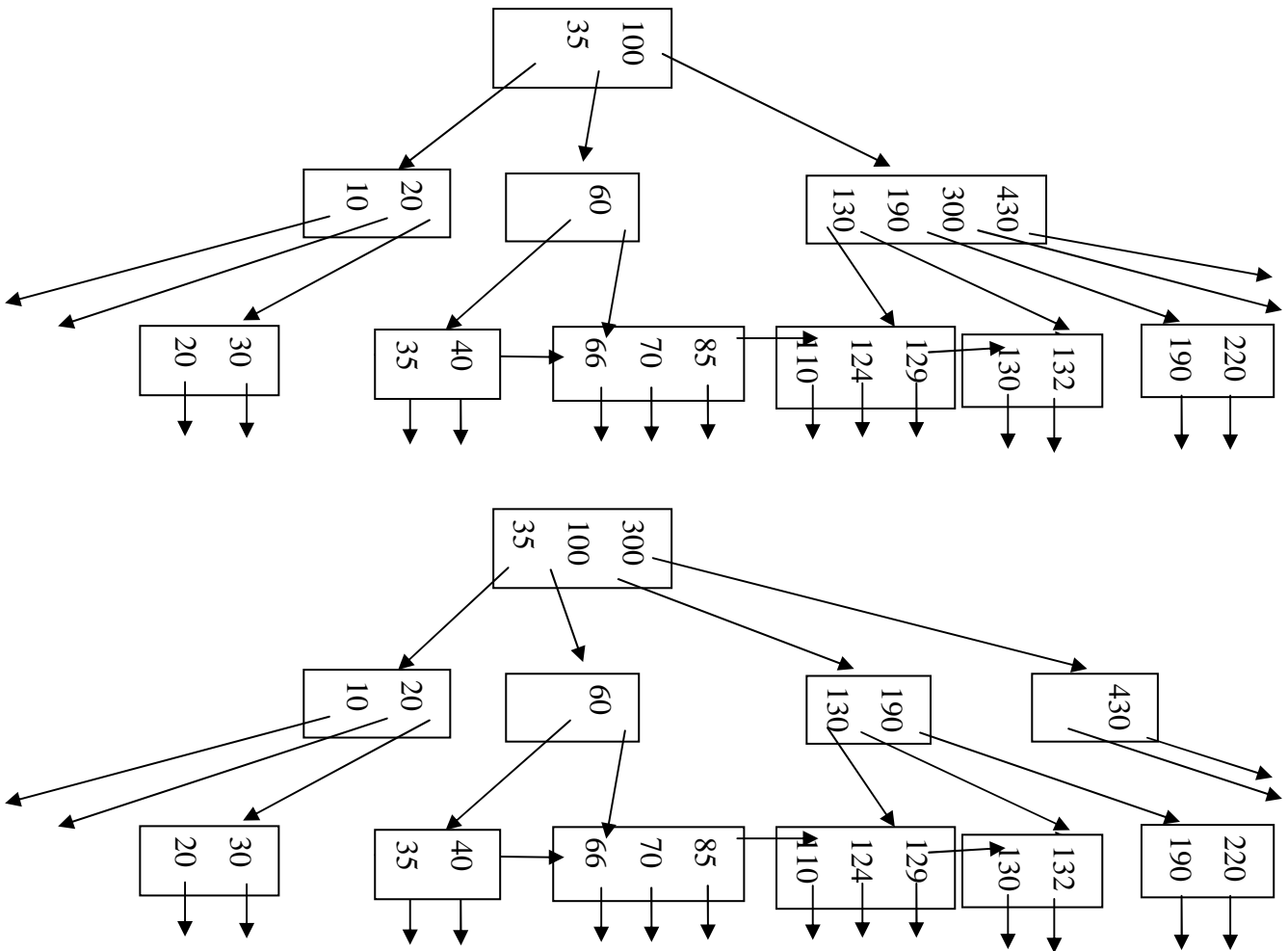
I due schedule sono view-equivalenti

Esercizio 3

inserimento delle chiavi 35 e 130







Esercizio 4

buffer di memoria centrale disponibili: $M=50000$
 dimensione del buffer: $B=4000$ bytes
 numero di tuple: $T(\text{Utente})= 800000$, $T(\text{Servizio})= 200000$,
 dimensione delle tuple: $S(\text{Utente})=500$ bytes, $S(\text{Servizio})=1600$,
 $S(\text{Fornitore})=100$
 valori: $V(\text{Utente, Fornitore})=4000$, $V(\text{Servizio, Fornitore})=20000$
 range Servizio.Costo= $[1,100]$

Calcoliamo innanzitutto il numero di blocchi occupati da ciascuna relazione:
 $B(\text{Utente})=T(\text{Utente}) \cdot S(\text{Utente})/B = 800000 \cdot 500/4000=100000$
 $B(\text{Servizio}) = 200000 \cdot 1600/4000=80000$

Prima sequenza: $(\sigma_{\text{Costo}>70} \text{ Servizio}) \triangleright \triangleleft \text{ Utente}$

Sia $X = \sigma_{\text{Costo}>70} \text{ Servizio}$
 C'è un indice secondario su Servizio.Costo
 $F=(100-70)/100-1+1=0.3$
 $\text{CostoIndice}(X) = 3 + f \cdot T(\text{Servizio}) = 3 + 0.3 \cdot 200000=60003$

$$B(X) = 0.3 * 80000 = 24000$$

X sta in memoria centrale quindi posso usare il join a un passo
Costo($X \bowtie U$) = $B(X) + B(U) = 24000 + 100000 = 124000$

$$\text{Costo totale} = 60003 + 124000 = 184003$$

Seconda sequenza: $\sigma_{\text{Costo} > 70}(\text{Servizio} \bowtie U)$

Sia $Z = \text{Servizio} \bowtie U$

Dato che sia Servizio che Utente sono ordinate su Fornitore, posso usare il Sort join.

$$\text{Costo}(Z) = B(\text{Servizio}) + B(U) = 100000 + 80000 = 180000$$

$$T(Z) = T(U) * T(\text{Servizio}) / \max\{V(U, \text{Fornitore}), V(\text{Servizio}, \text{Fornitore})\} = 800000 * 200000 / 20000 = 8000000$$

$$S(Z) = S(U) + S(\text{Servizio}) - S(\text{Fornitore}) = 500 + 1600 - 100 = 2000$$

$$B(Z) = 8000000 * 2000 / 4000 = 4000000$$

Dato che non ci sono indici su Z:

$$\text{Costo: } (\sigma_{\text{Costo} > 70} Z) = B(Z) = 4000000$$

$$\text{Costo totale} = 180000 + 4000000 = 4180000$$