2) • Block nested loop can always be used: for each block of pages of outer relation, read all pages of inner one.
Cost: $B(S) + B(R)[1 + B(S)/M] = 3000 + 6000(1 + 3000/100) = 3000 + 6000 \cdot 31 = 189000$

• Two pass algorithm can be used cause $6000 + 3000 \leq 102(102-1)$
Cost: $3(B(R) + B(S)) = 27000$

Generate $M(R)$ and $M(S)$ sublist such that $M(R) + M(S) = M$. Then read one page for each of them and join tuples

3) Index has 20 data entries per page $\longrightarrow$ $300000/20 = 15000$ pages $\longrightarrow$ 67% rule $= 22.500$ pages

300 tuples for each cost.

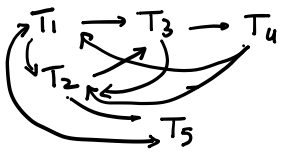
10 values $\longrightarrow$ $300 \cdot 10$ tuples $= 3000 \longrightarrow 300$ pages

$F = \frac{20+10}{2} = 15$

Cost $= \log_{15} 22500 + 1 + 300 = 305$ p.a.

4) • It is ACR cause every transaction reads from already committed transaction

• Precedence graph



Not conflict serializable cause p.graph is not acyclic

Yes if we delete T_2 and T_3

• $T_2 < T_5$ cause T_5 is last write of V
$T_2 < T_3$ cause T_2 need to read from default z
$T_1 < T_3$ //
$T_3 < T_4$ cause T_3 need to read from default x
$T_4 < T_1 < T_2$ cause T_1 need to read from $W_4(x)$
$T_4 < T_2$ cause T_2 is last write of x
$T_5 < T_1$ cause T_1 need to read from $W_5(y)$

Not possible $\longrightarrow$ Not view serializable

5) • $W_1(x)\ W_2(x)\ W_1(x)$
$T_2 \rightarrow T_1 \Longrightarrow$ view-equivalent for last write $\Longrightarrow$ view-serializable

• $W_1(x)\ W_2(x)\ W_1(x)\ W_3(x)$
Not accepted by 2PL but view-serializable ($T_1\ T_2\ T_3$ or $T_2\ T_1\ T_3$)