

2.1 Merge Algorithm

(a) Number of passes

Given $N = 12,000,000$ pages and $B = 60$ buffers, we can calculate the number of runs as follows:

$$k = \# \text{ of runs} = 1 + \lceil \log_{B-1}(\frac{N}{B}) \rceil = 1 + \lceil \log_{59}(\frac{12,000,000}{60}) \rceil = \boxed{4}$$

(b) Number of sorted runs

For first run, i.e. $i = 0$, we need to read N pages to generate $r_0 = \lceil \frac{N}{B} \rceil$ sorted runs.

For the remaining passes, where $i = 1, 2, 3, \dots, k - 1$, we need to read r_{i-1} runs to generate $r_i = \lceil \frac{1}{B-1} \times r_{i-1} \rceil$ sorted runs.

Therefore, the total number of sorted runs is:

$$\# \text{ of sorted runs} = r_0 + r_1 + r_2 = 200,000 + 3,390 + 58 = \boxed{203,448}$$

(c) Number of pages after Pass #2

Each initial run has $B = 60$ pages. After each run, the number of pages is multiplied by $B - 1 = 59$.

After Pass #1, each run is sorted and has $59 \times 60 = 3,540$ pages. Same, after Pass #2, each run has $59 \times 3,540 = \boxed{208,860}$ pages.

(d) Total I/Os

Given $N = 12,000,000$ pages and $B = 120$ buffers, we can calculate the total I/Os as follows:

$$k' = \# \text{ of passes} = 1 + \lceil \log_{B-1}(\frac{N}{B}) \rceil = 1 + \lceil \log_{119}(\frac{12,000,000}{120}) \rceil = 4$$
$$\text{Total I/Os} = 2N \times \# \text{ of passes} = 2N \times k' = 2 \times 12,000,000 \times 4 = \boxed{96,000,000}$$

(e) Smallest buffer size for 3 passes

Set buffer size b for 3 passes, given $N = 12,000,000$ pages, we have:

$$\begin{aligned} 3 &= 1 + \lceil \log_{b-1}(\frac{N}{b}) \rceil \\ 2 &= \lceil \log_{b-1}(\frac{N}{b}) \rceil \\ 1 &< \log_{b-1}(\frac{N}{b}) \leq 2 \\ b-1 &\leq \frac{N}{b} < (b-1)^2 \\ b &\geq 230 \end{aligned}$$

Therefore, the smallest buffer size for 3 passes is $\boxed{230}$.

(f) Maximum number of pages under 4 passes

Set number of pages p under 4 passes, so $N = p$ pages, given $B = 25$ buffers, we have:

$$\begin{aligned}4 &= 1 + \lceil \log_{B-1}(\frac{p}{B}) \rceil \\3 &= \lceil \log_{B-1}(\frac{p}{B}) \rceil \\2 &< \log_{B-1}(\frac{p}{B}) \leq 3 \\(B-1)^2 &\leq \frac{p}{B} < (B-1)^3 \\14,400 &< p \leq 345,600 \\p &\leq 345,600\end{aligned}$$

Therefore, the maximum number of pages under 4 passes is $\boxed{345,600}$.

2.2 Join Algorithm

(a) I/O cost for simple nested loop join

Denote total tuples in A and B as $m = M \times 50 = 3,000 \times 50 = 150,000$ and $n = N \times 100 = 800 \times 100 = 80,000$, we have:

$$\text{I/O cost} = M + (m \times N) = 3,000 + (150,000 \times 800) = \boxed{120,003,000}$$

(b) I/O cost for block nested loop join

$$\text{I/O cost} = O + (\lceil \frac{O}{F-2} \rceil \times N) = 2,000 + (\lceil \frac{2,000}{498} \rceil \times 800) = \boxed{6,000}$$

(c) I/O costs for sort-merge join

For sorting table B , we have:

$$\begin{aligned}\# \text{ of passes} &= 1 + \lceil \log_{F-1}(\frac{N}{F}) \rceil = 1 + \lceil \log_{499}(\frac{800}{500}) \rceil = 2 \\ \text{I/O cost} &= 2 \times N \times \# \text{ of passes} = 2 \times 800 \times 2 = \boxed{3,200}\end{aligned}$$

In the worst cases at merge phase, the attribute are same in two sorted runs, we need to compare all tuples in two runs. We have:

$$\text{I/O cost} = N + \lceil \frac{N}{F-2} \rceil \times O = 800 + \lceil \frac{800}{498} \rceil \times 2,000 = \boxed{4,800}$$

However, according to Tutorial 9: the I/O costs should be $= N \times O = 800 \times 2,000 = \boxed{1,600,000}$

(d) I/O costs for merge sorted runs

In the final merge phase, since there is no duplicate tuples, we only need to compare all tuples in two sorted runs in one pass. We have:

$$\text{I/O cost} = P + O = 600 + 2,000 = \boxed{2,600}$$

(e) I/O costs for hash join

Since there is no duplicate tuples, we only need to compare all tuples in two sorted runs in one pass.
We have:

$$\text{I/O cost for partition phase} = 2 \times (N + O) = 2 \times (800 + 2,000) = \boxed{5,600}$$

$$\text{I/O cost for probe phase} = N + O = 800 + 2,000 = \boxed{2,800}$$