Cost Estimation

- Based on:
  - operations
  - file and record sizes
  - file structures
  - block layout on disk

- Estimate:
  - number of seeks
  - number of rotational delays
  - number of blocks transferred (ignore unless > 10 or so)
  - (ignore memory-access time and CPU time)
Cost Estimation – Selection Example

\( \sigma_{\text{ArrivalDate}} = 15 \text{ May} \)
Assume: 10,000 records, 50/block

1. sequential file, not contiguous, not sorted
   
   \( 10,000/50 = 200 \) block accesses
   
   @ 12 ms per access = 2400 ms = 2.4 sec

2. sequential file, contiguous, not sorted

   With 132 blocks per track, 200 blocks fit on a cylinder; 1 access + 200 block transfers
   
   \( = 12 \text{ ms} + 200 \times 68 \mu\text{s} \)
   
   \( = 12 + 13.6 \text{ ms} = 25.6 \text{ ms} \)

3. sequential file, contiguous, sorted (on ArrivalDate)

   Binary search \((\log_2 200) = 8\); so
   
   1 seek + 8 rotational delays = \( 8 \) ms
   
   + \( 8 \times 4 \) ms = 40 ms (worse than unsorted)

CostEst: 2
Cost Estimation – Selection Example

\[ \sigma_{\text{ArrivalDate}} = 15 \text{ May} \]
Assume: 10,000 records, 50/block

4. sequential file, not contiguous, sorted (on ArrivalDate), indexed, with index in memory

If considerably fewer than 50, all records are likely to be in one block;
1 access = 12 ms.

5. ..., index not in memory

Probably need 200 value-pointer pairs for sparse index. If pointer size is 4 bytes and date size is 8, \(\frac{512}{12} = 42.6\), so say 40 pairs per block and thus 5 blocks (contiguous). Thus, 1 access for index + 1 access for record = 24 ms.

CostEst: 3
Cost Estimation – Selection Example

\[ \sigma_{\text{ArrivalDate}} = 15 \text{ May} \]
Assume: 10,000 records, 50/block

6. sequential file, not contiguous, not sorted (on ArrivalDate), but with secondary index in memory on ArrivalDate

   If we estimate, with the help of someone who knows the application, that the records we need are in 20 different blocks, we have 20 accesses = 240 ms.

7. …, contiguous, …

   For 20 blocks, we have 1 seek + 20 rotational delays
   \[ = 8 \text{ ms} + 20 \times 4 \text{ ms} = 88 \text{ ms}. \]
Cost Estimation – Join Example

$$\Pi_{\text{City}} \sigma_{\text{ArrivalDate} = 15 \text{ May}} (s \mid \times \mid g)$$

Assume: 10,000 @ 50/block for $s$ and 10,000 @ 25/block for $g$.

Assume: sequential files, not contiguous, $g$ is sorted and has an in-memory index on GuestNr.

1. simple nested-loop join (index not used).

   For each block of $s$ (200) access each block of $g$; $200 + 200 \times 400 = 80,200$ accesses $= 962,400 \text{ ms} = 16 \text{ min}$

2. for each block of $s$, use the index to access blocks of $g$

   At most 50 blocks of $g$ for each block of $s$; likely less, say 20; then $200 + 200 \times 20 = 4,200$ accesses $= 50,400 \text{ ms} = 50.4 \text{ sec.}$

CostEst: 5
Cost Estimation – Join Example

\[ \pi_{\text{City}} (\sigma_{\text{ArrivalDate} = 15 \text{ May}} \ s \mid \times \ g) \]
(equivalent optimized)

Assume: 10,000 @ 50/block for s and 10,000 @ 25/block for g.

Assume: sequential files, not contiguous, g is sorted and has an in-memory index on GuestNr.

3. Get 15-May records from s and do simple nested-loop join (index not used).

   Access each s block once to get 15-May GuestNr’s and then access each g block once; 200 + 400 = 600 accesses @ 12 ms = 7,200 ms = 7.2 sec

4. Get 15-May records from s and use index to do join.

   Assume we access, say, 10% of the blocks of g using the index; 200 + 40 = 240 accesses or 2,880 ms = 2.9 sec.

CostEst: 6