Neha Maddali

Homework 2

Suppose I have a relation Grades(student_id, assignment_id, score). I have 200 students and 20 assignments. I would grade all submissions of one assignment based on the submission order, and then insert the records. As a result, based on my insertion nature, the student_id is not sorted, but the assignment_id is. I choose heap file as my file organization. My page is quite small – it can only store 40 records, or 200 bytes in one page. The SearchKeySize is 2 bytes and PointerSize is 2 bytes. My buffer size is also small, 4 pages.

Number of records = 200 * 20 = 4,000 records
Number of pages = 4,000 / 40 = 100 pages
--> each index node can have 49 search keys and 50 pointers

1. (50 points) If my most frequent query is to find individual students, such as
   select * from grades where student_id='3347';
   a. (5pts) What is the I/O cost (i.e., number of pages in terms of reading and writing) for this query if I don't build index for student_id? (note: student_id can appear as many as 20 times in this relation)
      Need to read the entire relation: 0.5 * 10 (pages) * 20 (pages to read and write) = 100
   b. I want to improve the I/O cost. I am debating if I need to build index for student_id, or to sort based on student_id. So I need to do some estimation. Please help me by answering the following questions.
      i. (15pts) What is the I/O cost of multi-way merge sort (aka, external sort) if I sort the relation after I enter all records? Explain the process.
         Phase 1: partitioning. Reading = 100. Writing = 100. I/O = 200. Get 25 subfiles
         Phase 2:
            --> round 1. Merge every 3 subfiles into one
                Reading = 100. Writing = 100. I/O = 200. Get 9 subfiles.
            --> round 2. Merge every 3 subfiles into one
                Reading = 100. Writing = 100. I/O = 200. Get 3 subfiles.
            --> round 3. Merge every 3 subfiles into one
                Reading = 100. Writing = 100. I/O = 200. DONE.

         Total = 4 * 200 = 800

      ii. (15pts) Suppose I decide to build B+ tree index instead of sorting. What is the smallest number of pages do you estimate the B+ tree will take?
         one pointer per record – dense index
         4,000 pointers on the leaf level
         4,000 / 49 = 82 pages on the leaf level
         The tree is three level, 82 + 2 + 1 = 85
         Smallest number of pages estimated the B+ tree will take = 85 ± 2
      iii. (15pts) What is the worst I/O cost for answering this query with B+ tree index now?
         Read tree: 3 or 4

Read data: 20
-- > 20 + 3 = 23, 20 + 4 = 24
Worst I/O cost: 23 or 24

2. (40 points) If my most frequent query is to find all scores for an assignment, such as select * from grades where assignment_id='01';
    a. (10pts) What is the I/O cost if I don't build index for assignment_id? (note: assignment_id is sorted and each assignment_id can appear as many as 200 times in this relation)
    Binary search: $\log_2(100) = 6$ or 7 pages
    Retrieve all records: 200 / 40 = 5 pages
    6 + 5 = 11, 7 + 5 = 12
    I/O cost: 12 ± 2
    b. I am debating if building index for assignment_id would further improve the I/O cost. Please help me by answering the following questions.
        i. (15pts) Suppose I decide to build B+ tree index. What is the smallest number of pages do you estimate the B+ tree will take?
        One pointer per page – sparse index
        100 pointers on the leaf level
        100 / 49 = 3 pages on the leaf level
        The tree has two levels
        3 + 1 = 4
        ii. (15pts) What is the best I/O cost for answering this query with B+ tree index now?
        Read tree: 2 or 3
        Read data: 5 ± 1
        Best I/O: 6 to 9
3. (10 points) Suppose at the end of the semester, I need to curve the grades. I decide to increase all scores by 5 points. What is the I/O cost for this operation?
    Need to read and write the entire relation: 100 (read) + 100 (write) = 200

Submission Instruction

Do NOT handwrite. Submit all answers in a SINGLE file, in PDF format, through your Canvas account. Please explain your estimation for each question. You will get points deduction if you do not provide explanations.