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.

1. (50 points) If my most frequent query is to find individual students, such as  
   select \* from grades where student\_id=‘3347’;  
   select \* from grades where student\_id=‘8462’;
   1. (5pts) What is the I/O cost (i.e., number of pages in reading and writing) for finding one student if I don’t build index for student\_id? (note: student\_id can appear as many as 20 times in this relation)
      1. 200 students \* 20 assignments / 40(records/page) = 100 pages
   2. 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. **Buffer size 4**
      1. (15pts) What is the I/O cost of multi-way merge sort if I sort the relation after I enter all records?
         1. Sorted sub files: 4 pages each. 100 / 4 = 25 subfiles of size 4 pages.
            1. Read = 100, write = 100.
         2. 3 subfiles per merge: 8 subfiles of 4\*3=12 pages, last subfile is 4 pages
            1. Read = 100, write = 100.
         3. Repeat. 3 total subfiles, first 2 is 12\*3 = 36 pages, last is 12\*2 + 4 = 28
            1. Read = 100, write = 100.
         4. Final Merge: 3 subfiles into one single file.
            1. Read = 100, write = 100.
         5. I/O Cost = read + write = 4\*100 + 4\*100 = 800
      2. (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?
         1. SearchKeySize = 2, PointerSize = 2, pageSize = 200
         2. 2\*searchKeys + 2\*(searchKeys + 1) <= 200, searchKeys = 49.5
         3. 4000 records / 49 (searchKeys/page) = at least 82 pages
      3. (15pts) What is the worst I/O cost for answering those queries with B+ tree index now?
         1. Worst case: 1 + 1 + 20 hw’s/student = 22
2. (40 points) If my most frequent query is to find all scores for an assignment, such as  
   select score from grades where assignment\_id=‘01’;  
   select score from grades where assignment\_id=‘14’;
   1. (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)
      1. Data is sorted by assignment\_id.
      2. Assuming average case when there is a record for each homework for each student:
      3. We must read 200 (records/assignment\_id)/40(records/page = 5 pages/assignment
      4. Binary search cost = log2(100) + 5
   2. I am debating if building index for assignment\_id would further improve the I/O cost. Please help me by answering the following questions.
      1. (15pts) Suppose I decide to build B+ tree index. What is the smallest number of pages do you estimate the B+ tree will take?
         1. Need to store 100 searchKeys and pointers per page.
         2. Bytes needed = 2\*100 + 2\*100 = 400
         3. 400 (bytes/b+) / (200 bytes/page) = 2 pages/b+
      2. (15pts) What is the best I/O cost for answering those queries with B+ tree index now?
         1. 1 (r index) + 5 (r pages/assignment\_id) = 6
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?
   1. Read/Write whole file = 100 + 100 = 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.