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 record = 200\*20=4,000

Number of pages =4000/40=100

each index node can have 49 search key 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’;  
   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)

Need to read the entire relation: 100

If answer 100, full score  
If mention read entire relation, 4 points  
If mention read, 2 points  
If answer by notation (e.g.,0.5\*P\*D), and not mention read, 1 point

* 1. 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.
     1. (15pts) What is the I/O cost of multi-way merge sort if I sort the relation after I enter all records?   
          
        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  
        For each round/phase,  
        if answer is not 2 times Number of pages, -2 pts  
        if miss one, -3pts  
          
        total= 4\*200=800

if answer is not the sum of all reading+writing, -3 points  
(if they miss a round or have error in previous steps, but use all reading+writing, do not deduct points )  
  
if use formula (2N\*log…), and get different final results, -2 pts

* + 1. (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?   
         
       dense index, one pointer per record  
       4,000 pointers on the leaf level  
       4000/49 = 82 pages on the leaf level  
       the tree is three level  
       82+2+1=85  
         
       if answer is , full points  
       if answer is not around 85:  
       part 1, type of index, 5pts  
       if get 4000 pointers on the data entry (leaf) level, +5 points  
       if mention dense index or one pointer per record, but wrong number, +3 points



Part 2, number of leaf nodes, 5 pts  
if get that leaf level has pages, +5 points  
if trying to calculate leaf level pages, +3 points

Part 3, number of total nodes, 5pts  
if mention tree has 3 levels, +3 points

if mention tree has 1 or 2 levels, +1 points  
if sum up the number of nodes for each level of the tree, +2 points   
if sum up the number of nodes for each level of the tree (but miss one level), +1 points

* + 1. (15pts) What is the worst I/O cost for answering those queries with B+ tree index now?   
         
       read tree: 3 or 4  
       read data: 20  
       worst I/O, 23 or 24

If answer is 22 to 25, full points

if answer is not around 23 or 24:

if mention reading 3 or 4 index pages, +5 points

if mention reading index pages, +3 points  
if mention reading 20 data pages, +5 points  
if mention reading data pages, +3 points

if conduct the sum of reading index and data pages, +5 points

1. (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)  
      binary search: log\_2(100)= 6 or 7 pages

Retrieve all records: 200/40=5 pages

I/O cost = 7+5=12

If answer is , full score.

if answer is not around 12:

search, 4 points

if get search is 6 or 7 pages, +4 points

if mention binary search, +3 points  
if mention search, +2 points  
Retrieve, 4 points

if mention reading data pages, +4 points

if mention reading data pages, +3 points  
I/O cost, 2 points

if conduct the sum of search and reading data pages, +2 points

* 1. 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?   
          
        sparse index, one pointer per page  
        100 pointers on the leaf level  
        100/49 = 3 pages on the leaf level  
        the tree has two level  
        3+1=4  
          
        if answer is 3+1 or 2+1, full points  
        if answer is not 4:

part 1, type of index, 5pts  
if get 100 pointers on the data entry (leaf) level, +5 points  
if mention sparse index or one pointer per page, but wrong number, +3 points

Part 2, number of leaf nodes, 5 pts  
if get that leaf level has pages, +5 points  
if trying to calculate leaf level pages, +3 points

Part 3, number of total nodes, 5pts  
if mention tree has 2 levels, +3 points

if mention tree has 1 level, +1 points  
if sum up the number of nodes for each level of the tree (including root), +2 points   
if sum up the number of nodes for each level of the tree (but no root), +1 points

* + 1. (15pts) What is the best I/O cost for answering those queries with B+ tree index now?

read tree: 2 or 3  
read data:   
best I/O, [6,9]

If answer is 6 to 9, full points

if answer is not around 6 to 9:

if mention reading 2 or 3 index pages, +5 points

if mention reading index pages, +3 points  
if mention reading data pages, +5 points  
if mention reading data pages, +3 points

if conduct the sum of reading index and data pages, +5 points

1. (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 +100 =200  
if has read=100, +4 points  
if has read, +2 points  
if has write=100, +4 points  
if has write, +2 points  
if has read+write, +2 points

**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.