The deadline to upload your solution to iCorsi is on the 16th of May, at 23:59. Late submission policy will apply; see the Intro slides for details. Upload a PDF file containing your answers numbered according to the exercises. Write your name at the top of the PDF and name the file using the "HW5_FirstName_LastName" convention.

Description

This homework will give you experience in database physical design and computational costs associated with operations. Refer to the lecture "Physical Design And Query Execution Concepts".
The text in the solution **must not be handwritten** (e.g., you can use Word, LaTeX). Diagrams or small examples **can** be handwritten, photographed and pasted into the document. Consider also using drawing tools (e.g. diagram.net, LibreOffice draw). (Note: Handwritten diagrams or small examples must be clear.)

Exercises

Make suitable assumptions where necessary and state them explicitly in your answers.

A. Data structures

With respect to the following data structures used for indexing, describe scenarios when insertion:

- from a heap would take (a) 1 operation (b) N operations.
- from a sorted sequence would take log N + N operations.
- from a 2-3 tree would take log N operations.

B. Index and File

1. Explain with clear examples or diagrams what is a (a) dense index, and (b) sparse index. When is it advantageous to use a sparse index?
2. Explain with clear examples or diagrams what is (a) a clustered file (b) an unclustered file.
3. From among the four choices of building indexes we have seen in class (hint: Dense index clustered file, etc.), state with reasons the best and the worst choices.

C. B+ trees

Consider a disk containing 5 GiB (Gibibytes). The disk is mounted on a Linux server which uses a default physical block size of 256 B (Bytes). It will be used to access a 5 millions record inventory file, each record containing the following fields:

- ID: 8 bytes; no two records have the same value of ID.
- ItemName: 5 bytes
- Category: 4 bytes
- Quantity: 3 bytes

1. Using ID as the primary key, design a B^+ tree to serve as an index to this file by specifying the value of the parameter m as seen in class. What is the smallest and what is the largest number of children that an internal node can have?

2. Show schematically, as in Lecture 08 (slide 55), the narrow and the wide evolutions of the tree. What would be the number of leaves in the narrowest and the widest possible tree?

3. The user will be asking frequently questions about range of quantities, for example getting all the products below 200 units. To optimise these queries a secondary index needs to be created. Would you use a B^+ tree or a hash table? Why?

In all sub-questions, describe in detail how you decide to round non-integer values.

D. Unsorted join

Consider two unsorted tables R(A,B) and S(C,D) comprising of 2000 and 4000 blocks respectively.
Let us assume we have a RAM of 6 blocks, 2 of which are used for the allocation of R and 4 are used for S.
We need to compute the following query:

```
SELECT *
FROM R, S
WHERE R.B = S.C;
```

Count the total number of block reads needed to perform the query in the most efficient way, without sorting the data, i.e. without using merge join.