TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Alfons Kemper, Ph.D.

Exercise for *Database System Concepts for Non-Computer Scientist* im WiSe 19/20

Alexander van Renen (renen@in.tum.de)
http://db.in.tum.de/teaching/ws1920/DBSandere/?lang=en

Sheet 13

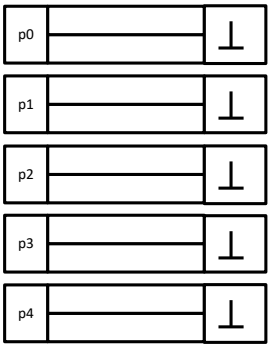
Exercise 1

[Same as in the previous sheet, but with a B+-Tree, instead of a B-Tree.] Calculate the optimal degree i and the number of required levels (also known as the "height" of the tree) for a B+-Tree with the following properties:

- The B+-Tree should store all humans currently living on earth (assume an even 10 billion).
- For each human we store the name, country and a unique identifier (100 Byte per human). The unique identifier will be used as the key an requires 8 Byte to store.
- The degree i of inner and leaf nodes may be different.
- Each node has to fit on a 16KB (16000 Byte) page.
- The page ids in the inner nodes require 8 Byte.
- This time (unlike in the lecture), we want to be precise: an inner node with n tuples requires $n + 1$ page ids to identify its children (in the lecture we simplifies this and assumed that a node with n tuples has n page ids).

Exercise 2

Please insert all tuples from the `Students` relation from the university schema into a hash table of size 5 (as in the figure). Each page can hold up to 2 tuples. As a means of handling collisions, linear chaining should be employed.



a) Use the following hash function: hash(key) = key mod 5.

b) Try using a better hash function: hash(key) = crc32(key) mod 5 To calculate the CRC32 of a key, you can use a website on the internet, for example: `https://crccalc.com/?crc=24002&method=crc32&datatype=ascii&outtype=dec`

Did the better hash function, result in a more evenly balanced hash table?