



CS660 Fall 2024 – Worksheet 1

Title: *External Sorting and Bitmap Indexes*

Q1. External Sort

Consider a file with $N=10,000$ pages. What is the smallest number of buffers B , so that external sorting will need only two passes? (Assume quicksort is used for in-memory sorting.)

Answer:

we want

$$B - 1 \geq \lceil N/B \rceil$$

so we know the solutions is close to $B_{sol} \approx \sqrt{N} = 100$

We plug in the numbers:

is $B = 100$ ok?

$$99 ? \lceil N/B \rceil = \lceil 10,000/100 \rceil = 100 - \text{not } \geq, \text{ so not that.}$$

is $B = 101$ ok?

$$100 ? \lceil N/B \rceil = \lceil 10,000/101 \rceil = 100 - \text{is indeed } \geq, \text{ yes.}$$

So correct answer is: **B = 101**

Q2. Bitmap Index.

Consider the following relation Instances:

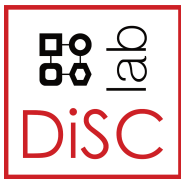
Instance ID	Storage Type	Storage Size (GB)	Memory Size (GiB)	vCPUs
OW88B1	SSD	475	122	16
3GI579	SSD	950	256	32
HCRNVN	NVMe SSD	475	64	16
INEX1V	NVMe SSD	1900	122	16
B15I2H	HDD	1900	256	64

1. Construct a bitmap index for the attributes `Storage Size (GB)` and `Memory Size (GiB)` for this table assuming all possible values (except **NULL**) for these two attributes are given in the above table, and the value of these two attributes are allowed to be **NULL**.

Storage Size (GB):

475: 10100, 950: 01000, 1900:00011, NULL: 00000

Memory Size (GiB):



64:00100, 122: 10010, 256: 01001, NULL: 00000

2. Assuming now we have N instances, how large space will the above two bitmap indexes occupy?

4N bits (N/2 bytes) for `Storage Size (GB)`, and 4N bits (N/2 bytes) for `Memory Size (GiB)`

3. Use the result from the above bitmap indexes and bitwise operations to answer the following query: give the total number of instances that have storage size larger than 800GB and have memory size as 256 GiB.

Typical bitwise operations include & (and), | (or), ~ (not), ^ (xor), >> x (right shift by x bits), << x (left shift by x bits). But we just need to select some of them that can be used here to answer the query.

Assuming the function `__builtin_popcount(x)` function in GCC can also be used here to count how many bits are set when the given number x is in a binary format.

`__builtin_popcount(((01000 | 00011) & 01001))`