

Adapted from Ex 8.4.2. In this problem, we consider indexes for the relation

`Ships(name, class, launched)`

from our running battleships exercise. Assume:

- i. `name` is the key.
- ii. The relation `Ships` is stored over 50 pages.
- iii. The relation is clustered on `class` so we expect that only one disk access is needed to find the ships of a given class.
- iv. On average, there are 5 ships of a class, and 25 ships launched in any given year.

- v. With probability p_1 the operation on this relation is a query of the form

`SELECT * FROM Ships WHERE name = n.`

- vi. With probability p_2 the operation on this relation is a query of the form

`SELECT * FROM Ships WHERE class = c.`

- vii. With probability p_3 the operation on this relation is a query of the form

`SELECT * FROM Ships WHERE launched = y.`

- viii. With probability $1 - p_1 - p_2 - p_3$ the operation on this relation is an insertion of a new tuple into `Ships`.

You can also make assumptions about accessing indexes and finding empty space for insertions that were made in Example 8.14.

- (a) If you can only create one index, how would you decide what index to create ?
- (b) If you can create any number of indexes, what are the possible index combinations ?
- (c) Consider the creation of indexes on `name`, `class`, and `launched`. For each combination of indexes, estimate the average cost of an operation. As a function of p_1, p_2 , and p_3 , what is the best choice of indexes ?