Exercises for DAT131 Complexity of Data Structures

You need to solve at least 4 of the 7 exercises below to pass the course.

Assignment 1. Consider the following data structure (which we call *initialised memory*). It maintains an array $A[1], \ldots, A[m]$ of numbers (let's say the numbers are between 1 and m, but that doesn't really matter for this exercise) and supports the following operations.

init: set $A[i] = 0$ for all $1 \leq i \leq n$.
write(i, x)**:** set $A[i] = x$
read(i)**:** return $A[i]$.

It is easy to implement this data structure so that read and write take constant time, and init takes $O(n)$.

Question Show how to implement all three operations in constant time, or prove that that is impossible. *Hint*: This is not a bit-fiddling exercise.

Assignment 2. Exercise 4 on Husfeldt's lecture notes

Assignment 3. Exercise 5 on Husfeldt's lecture notes

Assignment 4. Exercise 6 on Husfeldt's lecture notes

Assignment 5. Exercise 9b and 9c on Husfeldt's lecture notes

Assignment 6. Consider section 4.3 of Fich's notes for CSC2429 (Toronto), where the pseudocode for **insert** and **pred** for a van Emde Boas Tree is given. Write **delete**(x, S).

Assignment 7. A pseudo-implicit dictionary (cf. section 1 of Husfeldt's notes for notation) stores only members of S and a single other symbol, i.e., for all $1 \leq i \leq$ size,

$$M(i) \in S \cup \{\perp\}.$$

Describe a pseudo-implicit static dictionary with one probe and one-sided error ϵ, i.e.,

1. if $x \in S$ then the structure returns 'yes' after one probe,
2. if $x \notin S$ then the structure returns 'yes' with probability at most ϵ after one probe.

You may use space size $= O(n \log m)$.

Alternatively, show that such a structure cannot exist.