

Name:
ID:

CSCI 3104, Algorithms
Explain-It-Back 4

Profs. Grochow & Layer
Spring 2019, CU-Boulder

The ecology department is planning a large-scale fish migration study that involves electronically tagging and releasing millions of fish across North America, waiting six months, then trapping the fish and recording the where they found each fish, according to its tracking number. In previous smaller-scale experiments, the field scientists used a hand-held device that had a sensor for reading the electronic sensor and a small onboard hard drive that used a predefined table for storing the tag ID, timestamp, and current GPS. In this table, every possible tag had a preset row which allowed for very fast (constant-time) insertions and lookups. While the team would like to re-use this hardware, they do not think that there is enough hard drive space to account for a table with millions of rows. Help them figure out another solution that provides fast insertions and lookups without requiring large memory allocations. HINT: an individual scientist will only tag a few thousand fish at a time.

Dear Ecology Department,

I can see that you want to preserve the use of your hardware while maintaining the efficiency of accessing and inputting your data. The biggest problem that you have here is optimizing your hard drive space because if you know you are about to increase your data size you need to rethink your data structure. //

The biggest fix is to change how you are storing data. Right now you are allocating and storing data where you really shouldn't be. Since not every single fish that you tag will be caught your table will have a lot of unused space. To fix this we would implement a dynamically allocated hash table that will organize your data in a way that optimizes the allocation and storage. Basically, what this new structure will do is take your data and store it in a list which is ordered by a "look-up" value. This will decrease the amount of time it takes you to locate a single data point and in turn decrease the amount of time that it takes to allocate your data.

To help you picture what I mean by this think of a massive kitchen that is going to prepare a massive dinner at the Oscars. Every plate has a piece of paper that has information of what famous person the plate belongs to and what food will be placed on it. You want to stack all of these plates so that as you make food you want have an efficient plate-stacking-structure-order in which you can quickly locate the desired plate and place the food on it to

Name:
ID:

CSCI 3104, Algorithms
Explain-It-Back 4

Profs. Grochow & Layer
Spring 2019, CU-Boulder

serve. You can think of the individual stacks of plates as the lists of data that are grouped together by a "look-up" key, and the relating "look-up" key for the plates would that the plates are organized by table. This will drastically decrease your time for finding a plate because you can search initially by table, find the stack of plates and then just look through that stack, as opposed to search through every single plate every single time. Say that the guest list is ever growing (i.e. your data of fish still keeps coming in) you can easily add a plate to the correctly designated table just by placing the plate at the top of the pile. As you can see adding data will be very efficient as will searching through the lists of data.

There is a catch-22 here you have to maintain a balance of how many groupings of plates you have and how high you allow the plates to stack. The average number of the plates in the stack need to be equal to $\log(\text{of groupings of plates})$. This is very complicated to explain but there is plenty of science behind proving the efficiency of it.

You may ask, What happens if the same customer comes in, orders and then orders another plate? Well my good fellow, you would always keep that plate in the same location so that each time his order does come up you maintain the same location of it (wash the plate of course) so you don't have to go through the same process of re-inserting the plate back into the massive piles of plates.

Another cool thing this structure allows you to do is keep track the amount of times the same person orders their plate. This is in direct parallel to how many times you come in contact with the same fish. Think of it as putting a tally on the bottom of the plate, so if you ever want to see which of your customers comes in the most, all you have to do is find the plate, and look underneath for the tally stored on the bottom side. I hope you can draw the parallels that I was trying to get across here so if you do have any trouble, have questions, or if you want me to come over and implement this please feel free to contact me!

Best of luck! -Keaton Whitehead