

U173SEFUN : <@U5JUDH2UE> for assoc'ing a value if not null, with specter it's `(setval [:a some?] my-value amap)`

U5ZAJ15P0 : <@U0E0XL064> are those basically lenses?

U173SEFUN : for assoc'ing only if field exists, it would be `(setval (must :a) my-val amap)`

U173SEFUN : will be way more performant than merge as well

U0E0XL064 : I didn't use specter much yet - only found about it about a month ago - but it's definitely cool. Also for small cases.

U064X3EF3 : This is discussed in my talk, but if order matters, then your system is already broken. That is, you should never have two different versions of the same class on your classpath - if you do, then something bad has already happened.

U050MP39D : you're right, log32n. so if you allowed delete you'd have to rebalance the tree right? honestly I haven't thought about data structure algorithmic performance since uni, but I just kind of trust that this probably isn't feasible. I'm sure the fact that the data structures are persistent makes it even more complicated

U3JURM9B6 : disclaimer: I haven't read the actual source code

so now we both agree that a persistent vector is a tree with depth log32n

this means that assoc is NOT O(1), but O(log<sub>32</sub> n)

the reason being, when we update a node, we have to update "all ancestors" until we get to the root

U3JURM9B6 : since vectors are persistent => assoc has to create a new node => but then it has to create a new block for every ancestor from the thing we want to update until we get up to the root

U3JURM9B6 : so then the question is: can we do delete! in O(log<sub>32</sub> n) time, while making it easy to index -- and I don't know, but I'm leaning towards yes

U0561MZLJ : Style question: say I have three items `(def a ["a"]) (def b [{"b" "c"} {"d" "e"}]) (def c "f")`, and I want to put them into a list like so:``

`[~a ~@b ~c]

``

but without using syntax-quote/unquote/splicing-unquote...how would you do it as concisely as possible (without losing generality)?

U0561MZLJ : (i.e. the desired result is `[["a"] ["b" "c"] ["d" "e"] "f"]`)

U11BV7MTK : so you just want to append them?

U051SS2EU : the middle one is unpacked

U051SS2EU : one option is `(concat [a] b [c])` - I admit that looks weird

U0561MZLJ : <@U051SS2EU> actually, that's not nearly as bad as what I was doing

U0561MZLJ : Does suck a bit that you have to pack the elements you don't want unpacked just to...well...yeah

U051SS2EU : <@U0561MZLJ> depending on your actual problem, you could use something like `(def prepacked [x] (if (and (coll? x) (vector? (first x)))) x [x])` `(mapcat prepacked [a b c])`

U0561MZLJ : hah...yeah, that probably won't work because some of these are multiple levels nested, and only some of the levels need to be unpacked

U0561MZLJ : my personal preference is to "say what you mean" with syntax-quote, but understandably the team is worried about maintainability...

U14DSHZ4P : Hey there. How to update a key in a map to hold a empty vector?

U14DSHZ4P : ``=> (def Person {:person-id "person-1" :category "customer" :purchase ["p1"] :dates ["d1"]})#user/Person

=> (update :purchase Person [])

IllegalArgumentException Key must be integer clojure.lang.APersistentVector.invoke (APersistentVector.java:292)``

U0NCTKEV8 : look at the docs for update

U0NCTKEV8 : (you seem to have picked random arguments in a random order)

U0NCTKEV8 : (and you want assoc not update, but go ahead and read the docs for both assoc and update)

U0NCTKEV8 : by docs I meant the docstring you access using `doc` in the repl, but you may also want to start here

<[https://clojure.org/reference/data\\_structures#Maps](https://clojure.org/reference/data_structures#Maps)>

U5ZAJ15P0 : Hi! I am getting a cryptic error which I do not understand. Here's the piece of code:

<<https://gist.github.com/hmaurer/8e786bfd507798393c8be45ffb3a1b46>>. The error is in the comments. Could someone take a look please?

U5ZAJ15P0 : The error is `Can't let qualified name`, but I don't get why the gensym'ed variable gets qualified by the syntax quote

U14DSHZ4P : dumbol i was using the wrong function here. Thanks <@U0NCTKEV8>

U0NCTKEV8 : I would double check to make sure the code that is being run actually matches the code you are

reading (restart your jvms) and then I would suspect this is actually coming from some other macro  
U5ZAJ15P0 : I restarted lein's repl. Is that enough?