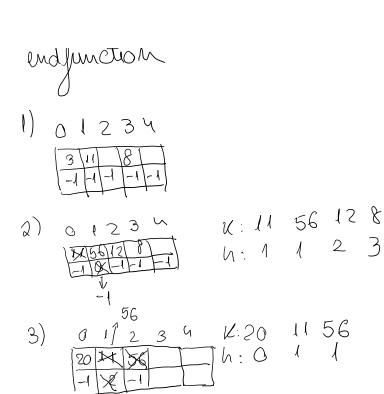
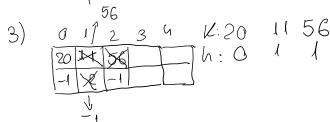
Seminar 06 Thursday, May 19, 2022 2:52 PM
Hashtables
Coolesced chaining
Implementation: Map - we will use only the keys (natural numbers).
Table [T] [] T: TElem
next the integer
first Free integer hash function (mod m (capacity)) Trunction
T: 18 13 15 16 31 5 26 10 11 12 Next: 1 4 -1 -1 6 0 -1 -1 -1 -1 -1
first Free = Ø 1467
K: 5, 18, 16, 15, 13, 31, 26
h(k):5,5,3,2,0,5,0
K%M
K%13
Map
function search (map, k) is:
$pos \leftarrow map.h(k)$
while map. Tipos] = K 1 pos = -1 execute
function search (map, k) is: pos <- map. h(k) while map. Tipos] # K 1 pos #-1 execute pos <- map. next [pos]

endwhile

if pos = -1 then

J search = false
else
else
and search = true





To remove Key K:

- juid the position where K is (P);

-search forward for an element that hashes at p;
- if we find an element we move backwards then we remove the

element; - otherwise we remove the element and set the links.

subalgorithm remove (map, R) is:

pos < map. h(K) prev < - 1

while post-11 map. 1405 J = K execus. prev'z pos pos = map. next[pos] endrible @ Key does not exist else index pos while index = -1 1 map. h (map. T Lindex J)!=pas execute: index = map. next [index] endwhile if index = - 1 then map. Tipos] - NULL_TELEM in map first Free < pos then map first Free < pos if prev + - 1 then map. next cprevs - map. next cpos] mop . next cps J < -- 1 else i = 0 while i < map. m 1 map. next [i] *pos execute enderile if i < map. in then V map. next cprev = map. next cpos] map. next cps J < --- 1 endif

else