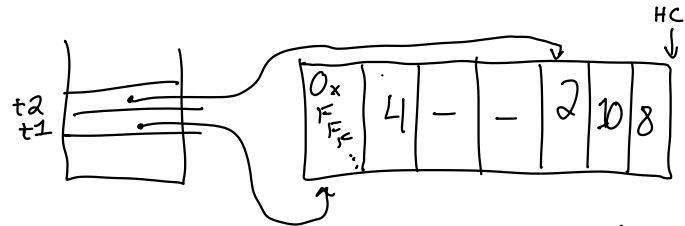


# Manual Memory Management

Consider Eagle extension

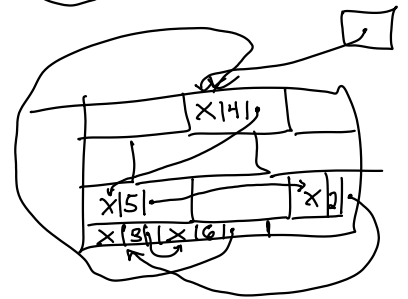
$\langle \text{expr} \rangle ::= \dots$   
 $\quad | \text{free}(\langle \text{expr} \rangle)$

let  $t1 = (2, 3, 4)$  in  
let  $t2 = (t1[0] + t1[1], t1[2])$  in  
let  $\text{junk} = \text{free}(t1)$  in  
 $t2$



A. Compact after every free

- $O(n)$  time,  $n = \text{size of heap}$
- still moves values around (so still need to rd ptrs)



B. Binary heap req for "free block"

- Free:  $O(1)$  time
- Alloc:  $O(n)$  time or keep metadata: where free memory is

# Malloc

- Initially,
1. Heap is composed of one "free block record" FBR
  2. Global ptr to some FBR
  3. FBRs form a cycle
  4. On allocation, walk chain to find big enough block (consider breaking it up)
  5. On free, create FBR in that space; add it to chain (consider merging)

