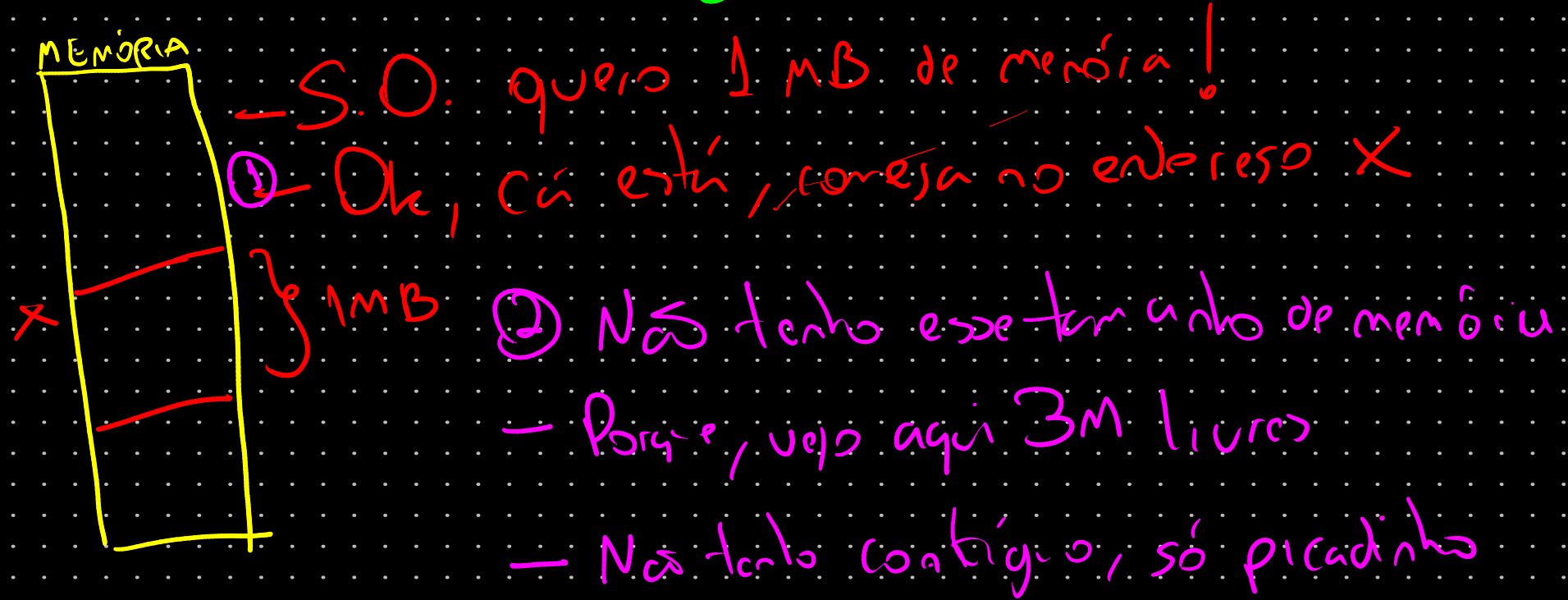


Listas Encadeadas (Linked Lists)

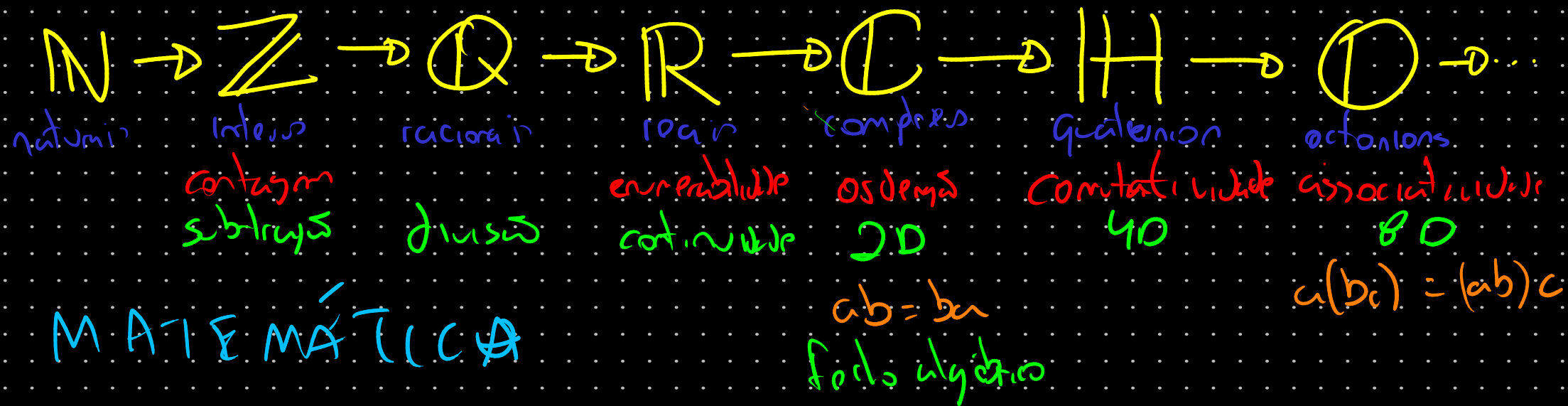
↳ Listas lineares homogêneas não-contíguas



— Ok, Save!



Ganhamos em utilização de espaço perdemos a indexação



NA COMPUTAÇÃO OCORRE O MESMO

Ao colocarmos uma propriedade, perde-se outra

Ex vetor \rightarrow Lista encadeada \rightarrow Árvore

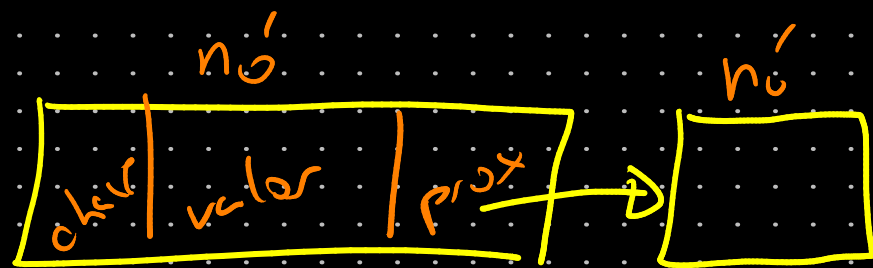
 indexação linearidade
 versatilidade complexidade menor

estrutura nó {

chave

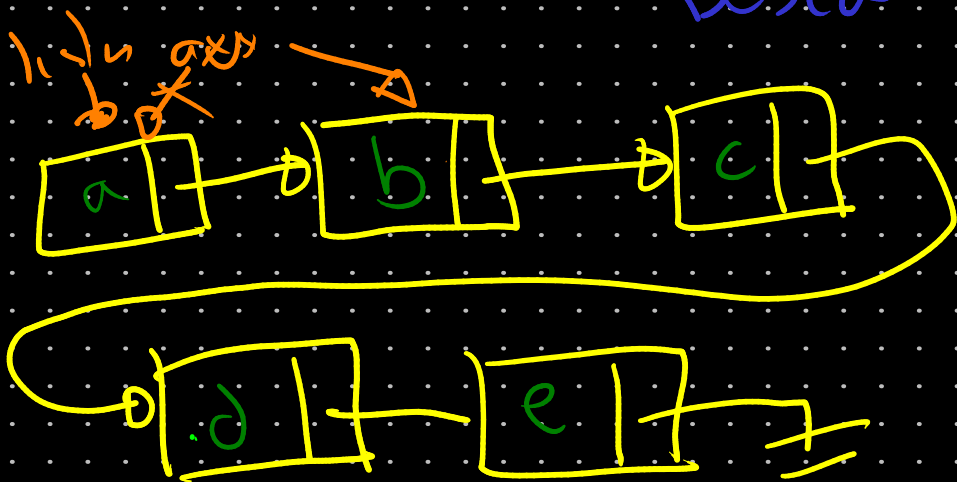
valor

prox (ponteiro para outro nó)



Operações {

- Insertar
- Remover
- Buscar



Algoritmo buscaLista(x, lista) {

- aux ← lista
- enquanto aux ≠ NULO ^{n vezes} faça
- se aux.chave = x então
- retorna aux
- senão
- aux ← aux.prox
- fim
- fin enquanto
- "Não encontrado"
- fim Algoritmo

$O(n)$

Insersão

insereLista(x, lista){

0(1) ✓ se lista = NULO então

0(1) lista ← criaNo() // alocação de memória

0(1) lista.chave ← x

0(1) lista próx ← NULO

senão ①

0(1) ① aux ← criaNo()

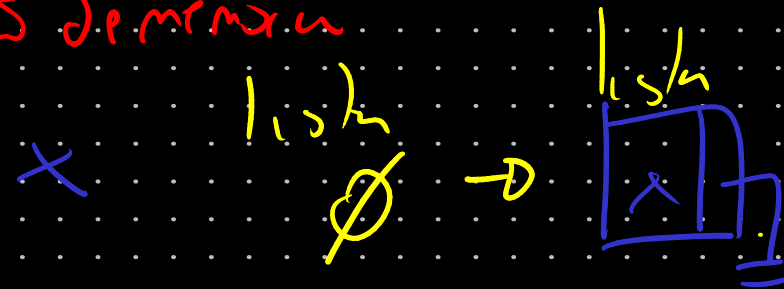
0(1) ② aux.chave ← x

0(1) ③ aux.prox ← lista

lista ← aux

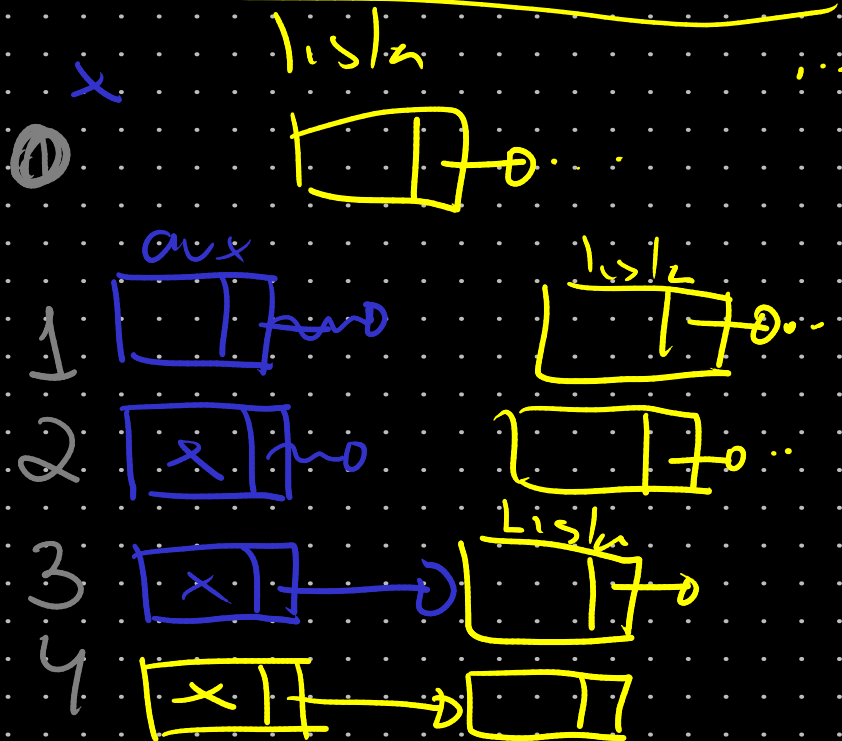
fim

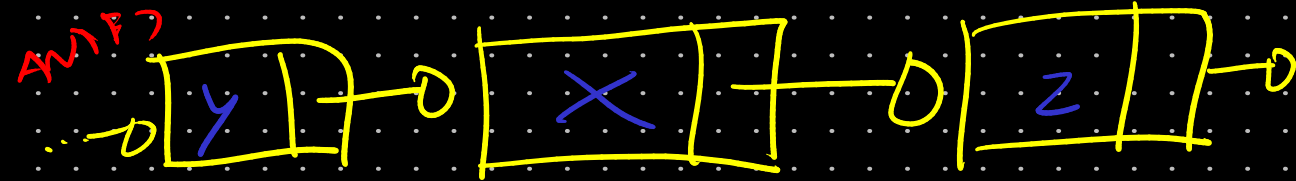
Função Algoritmo



$O(1)$

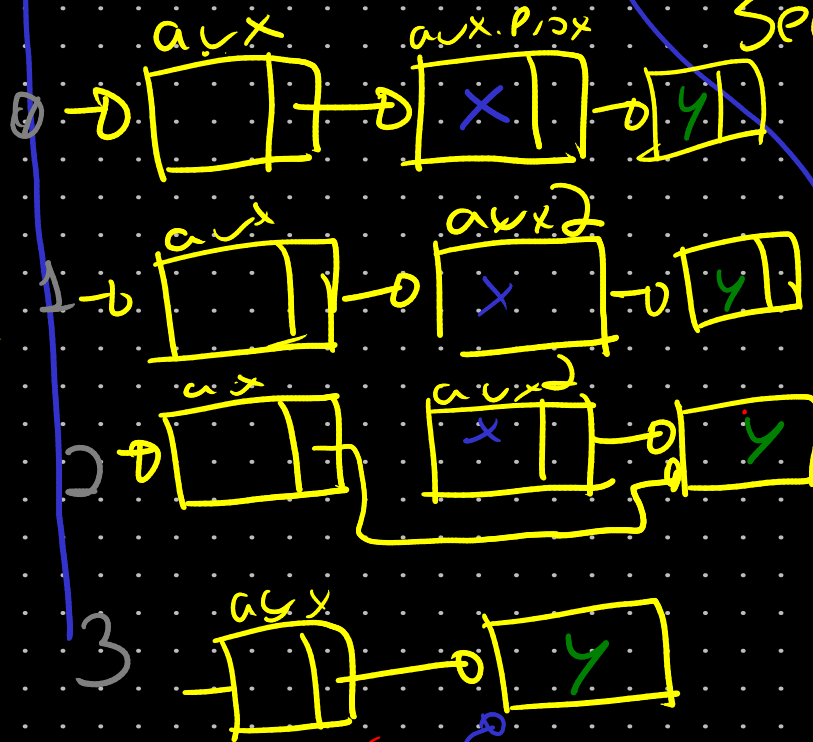
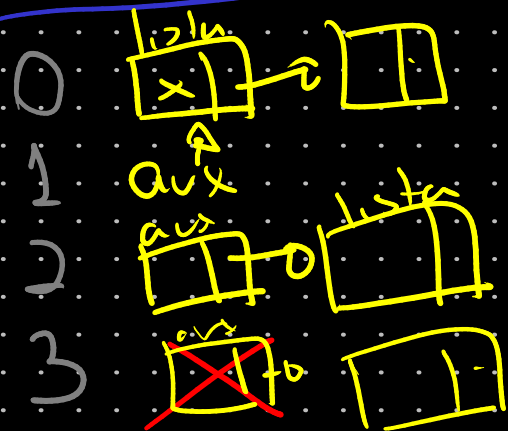
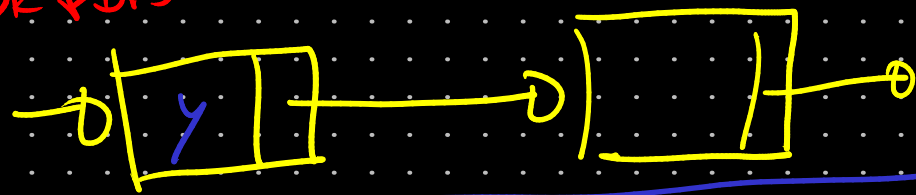
tempo constante





REMOÇÃO

DEPOIS



remover (x, lista) {
 se lista.chave = x então
 0 aux ← lista
 1 lista ← lista.prox
 2 libera (aux)
 3

Senão

aux ← lista

enquanto aux.prox ≠ nulo

0 se aux.prox.chave = x então

1 aux2 ← aux.prox

2 aux ← aux2.prox

3 libera (aux2)

return

Senão

aux ← aux.prox

fim enquanto
 fim

