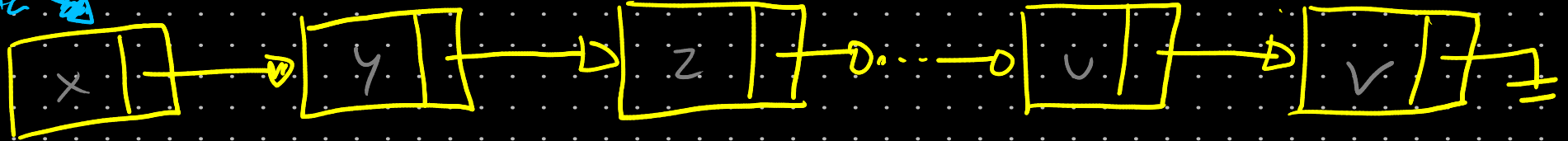


# Listas encadeadas (contínua)

Simple

lista

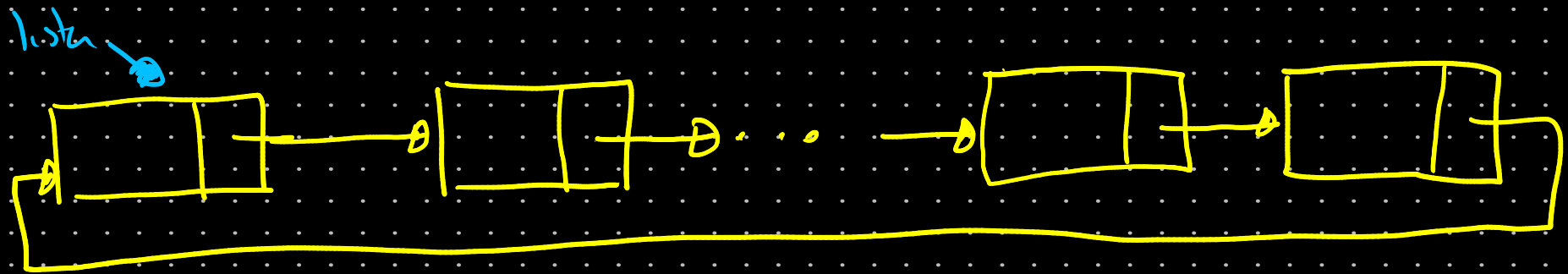
precisamos sempre manter uma variável indicando o início da lista



Para minimizar esse inconveniente, por vezes é interessante usarmos uma variável da lista encadeada.

x  
y  
z  
...

# 1) Lista encadeada circular



Interessante quando a ordem não importa.

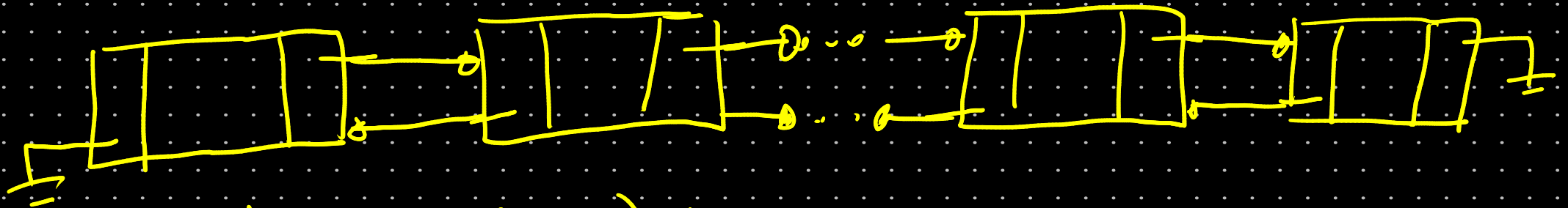
- Apontando para qualquer elemento da lista, conseguimos atingir qualquer outro elemento

\* Estabelecer um critério de parada ao "varrer" a lista  
percorrer



Lista com  
1 elemento

# D) Lista duplamente encadeada



Inclusão Lista Dupla (x, p) {

1)  $\begin{cases} \text{aux} \leftarrow \text{cnnovo}() \\ \text{aux.val} \leftarrow x \end{cases}$

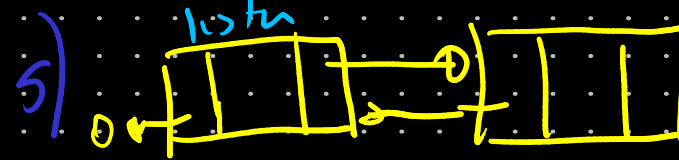
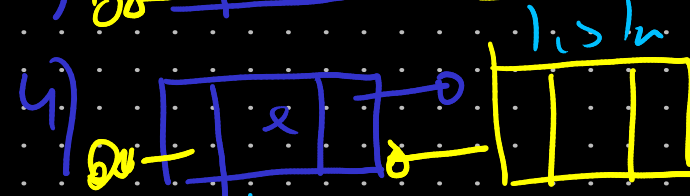
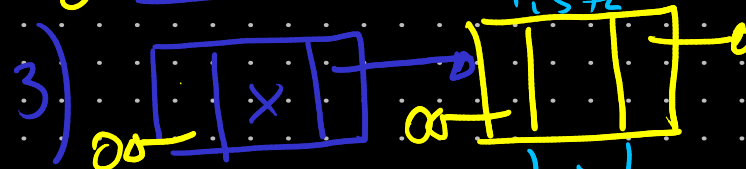
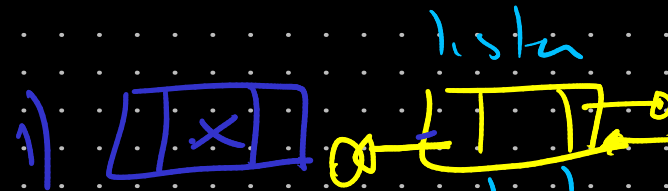
2)  $\text{aux.ante} \leftarrow p.\text{ant}$

3)  $\text{aux.prox} \leftarrow p$

4)  $p.\text{ant} \leftarrow \text{aux}$

5)  $p \leftarrow \text{aux}$

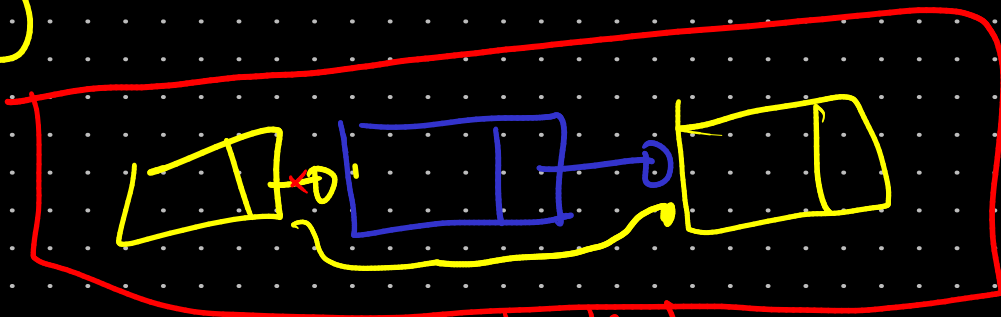
3)  $\uparrow$  lista ordenada  
é importante



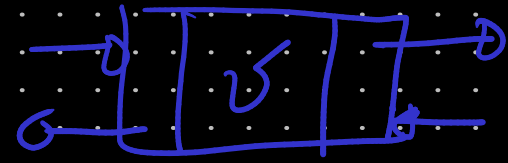
Remoção de Nó Encadeado (nó) {

- 1) { ant ← nó, ant  
próx ← nó.próx
- 2) { se ant ≠ NULO então {  
| ant.próx ← próximo
- 3) { se próximo ≠ NULO então {  
| próximo.ant ← ant
- 4) libera(nó)

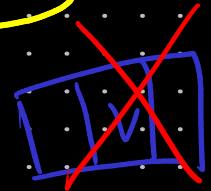
↑ qd a ordem  
é importante



Impossível de fazer  
com apenas a informação do nó



- 1) ant → nó → próximo
- 2) ant → próximo
- 3) próximo → ant
- 4) ant → próximo



# Complexidade

	Perda	Busca	Inserção	Ins Del	Remoção	Remoção
Simple	Em relação ao topo Contiguidade	$O(n)$	$O(n)$	$O(n)$	$O(n)$	X
Circular	Limite	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Dupla	Espaco (Memória)	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Dupla Circular	Limite / Espaço	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(1)$
	Ganho Utilidade					

Remoção Pontual de  
um nó apenas com  
essa informação



