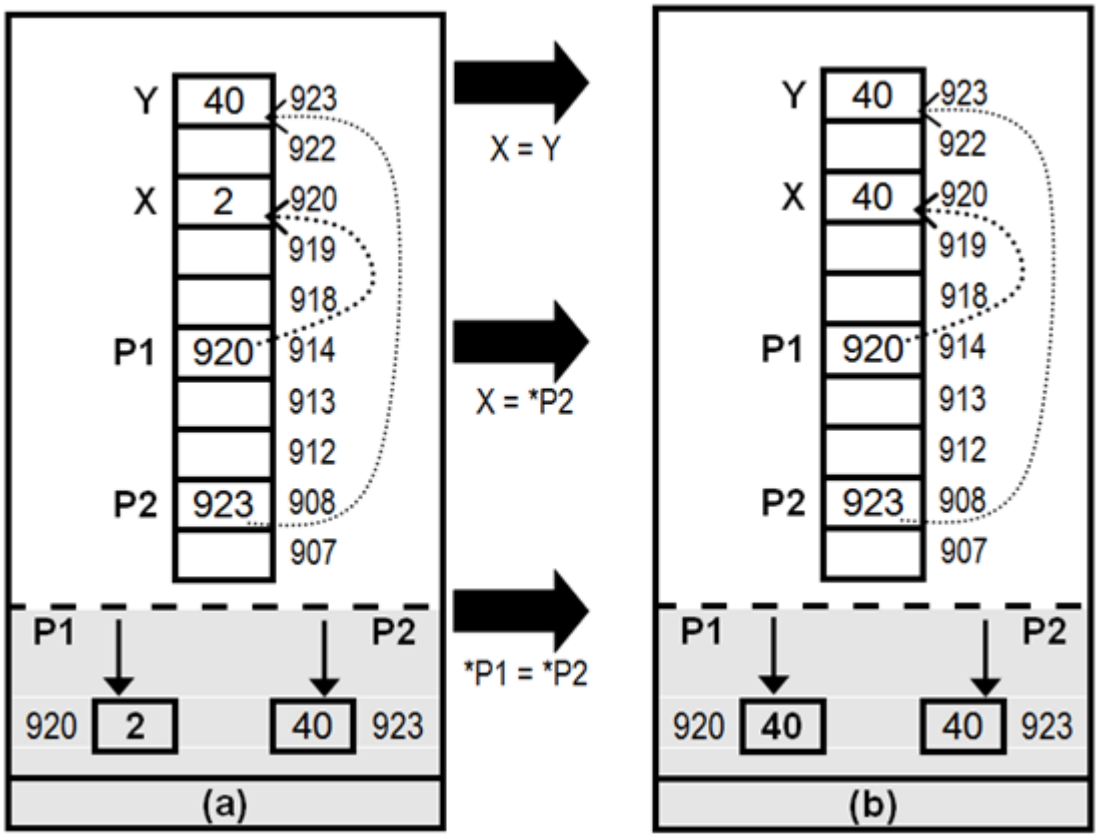
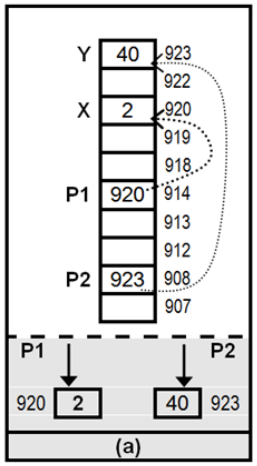
Estruturas de Dados com Jogos



Capítulo 5

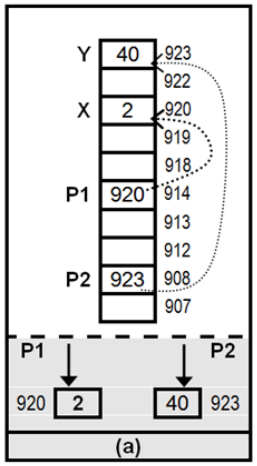
Listas Encadeadas com Alocação Dinâmica

Seus Objetivos neste Capítulo



- Entender o que é Alocação Dinâmica de Memória, no contexto do armazenamento temporário de conjuntos de elementos;
- Entender que a Alocação Encadeada e a Alocação Dinâmica são conceitos independentes que, quando combinados, formam uma técnica flexível e poderosa;
- Desenvolver habilidade para implementar estruturas encadeadas, com Alocação Dinâmica de Memória;
- Fazer uma reflexão visando escolher a técnica de armazenamento mais adequada aos jogos que você está desenvolvendo.

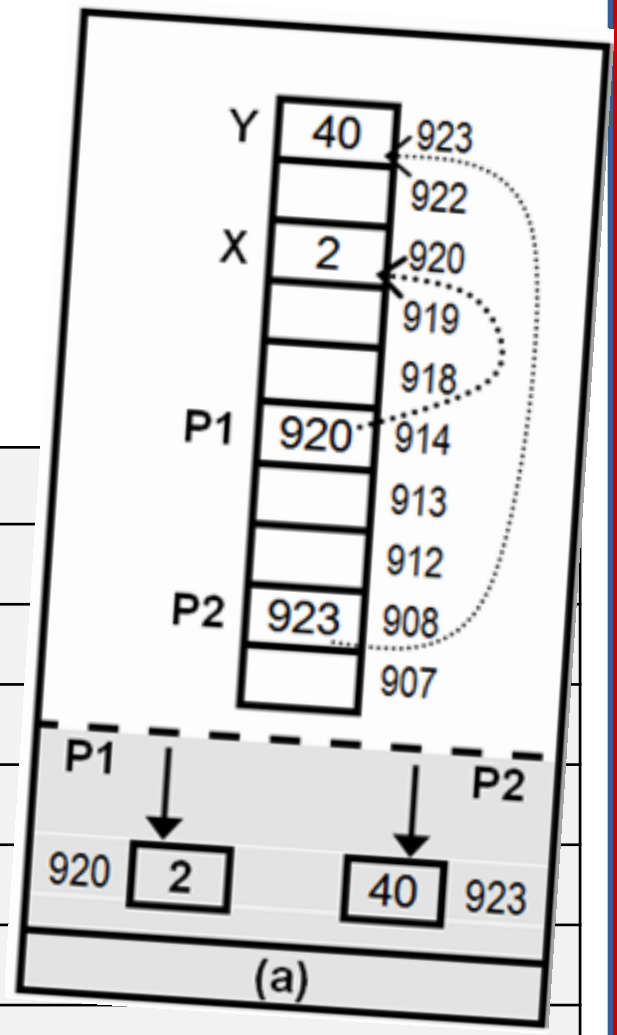
Na Alocação Dinâmica de Memória para um Conjunto de Elementos:



- Espaços de memória podem ser alocados no decorrer da execução do programa, quando forem efetivamente necessários;
- É possível alocar espaço para um elemento de cada vez;

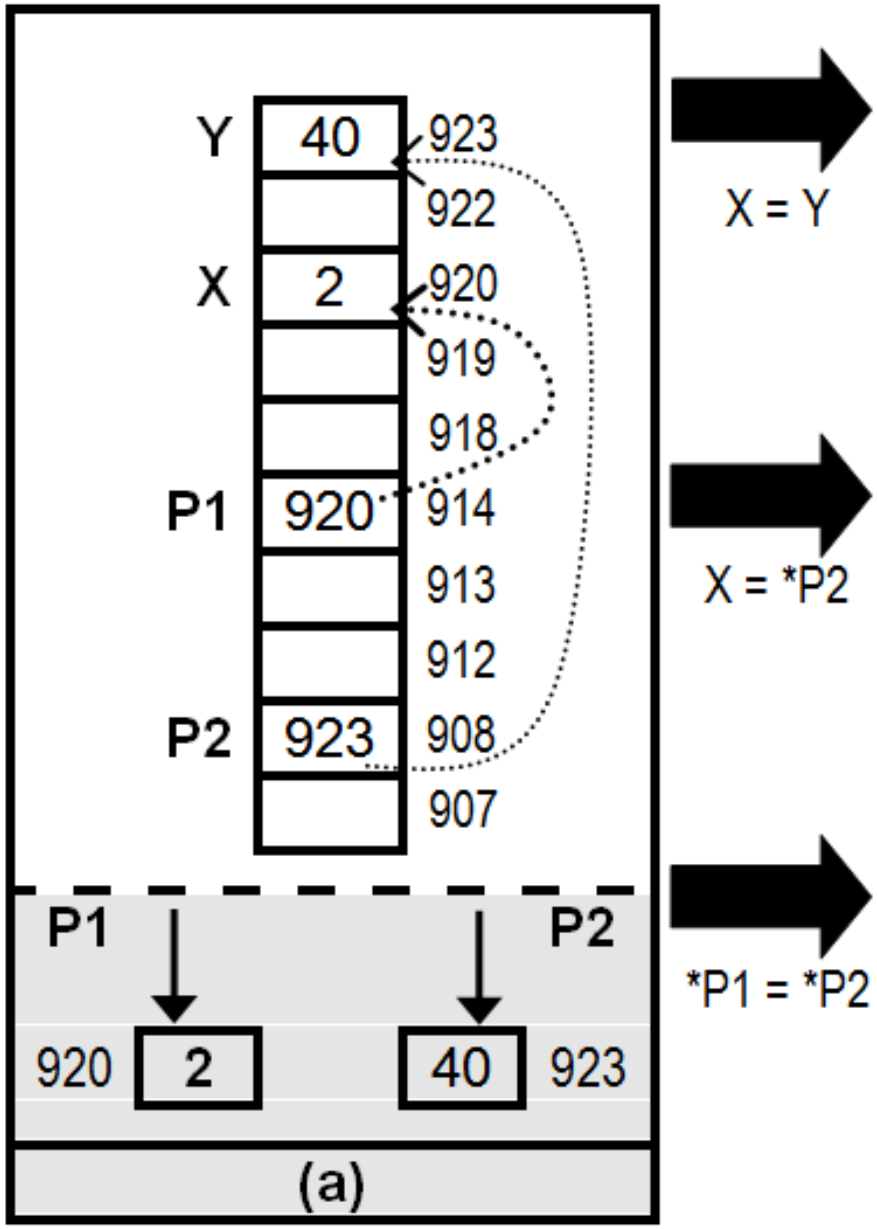
- Espaços de memória também podem ser liberados no decorrer a execução do programa, quando não forem mais necessários;
- Também é possível liberar espaço de um elemento de cada vez.

Alocação Dinâmica nas Linguagens C e C++

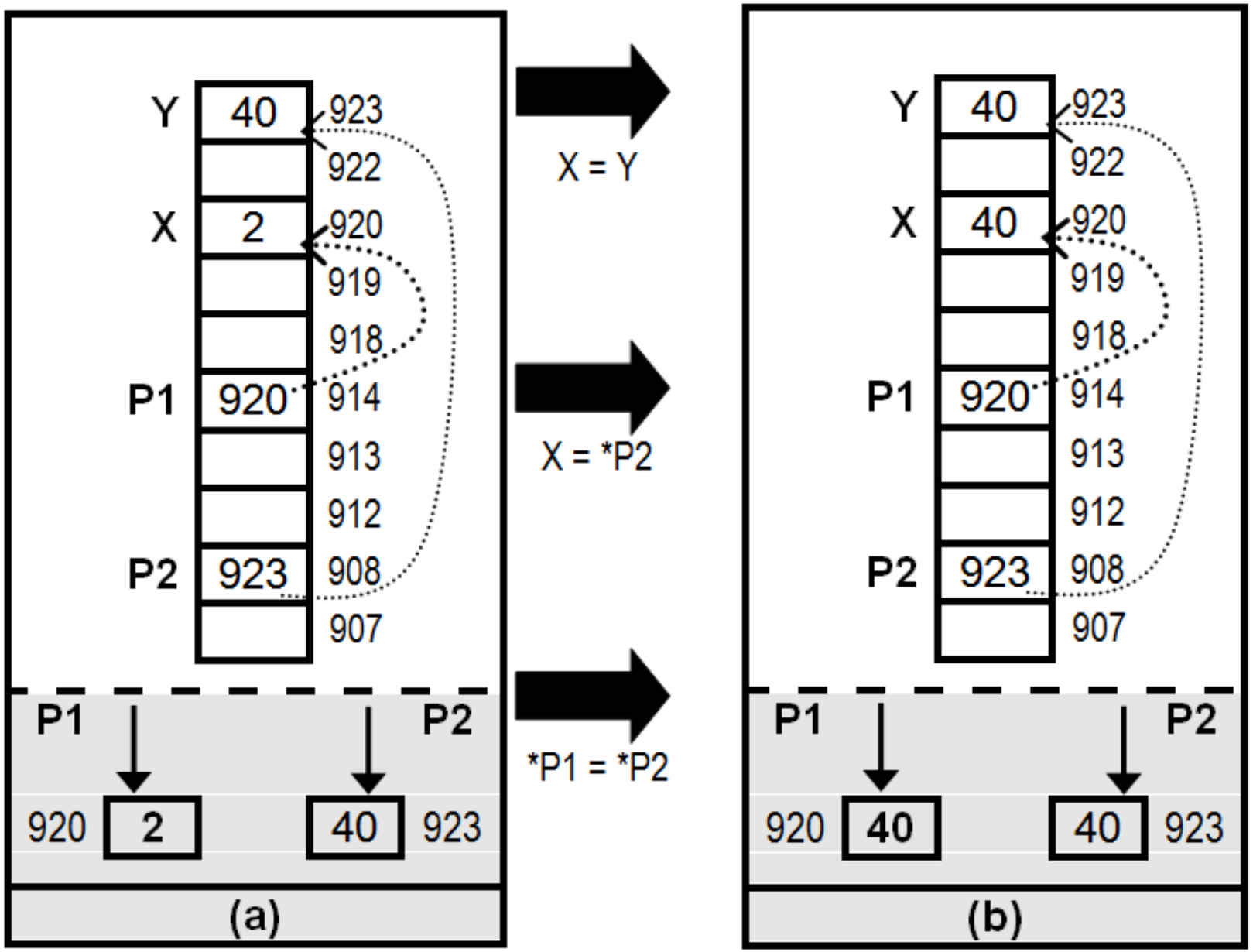


	C++	C
1	int X, Y;	int X, Y;
2	int *P1, *P2;	int *P1, *P2;
3	X = Y;	X = Y;
4	X = *P2;	X = *P2;
5	*P1 = *P2;	*P1 = *P2;
6	P1 = P2;	P1 = P2;
7	P1 = &X;	P1 = &X;
8	P1 = new int;	P1 = (int *) malloc((unsigned) (sizeof(int)));
9	delete P1;	free((char *) P1);

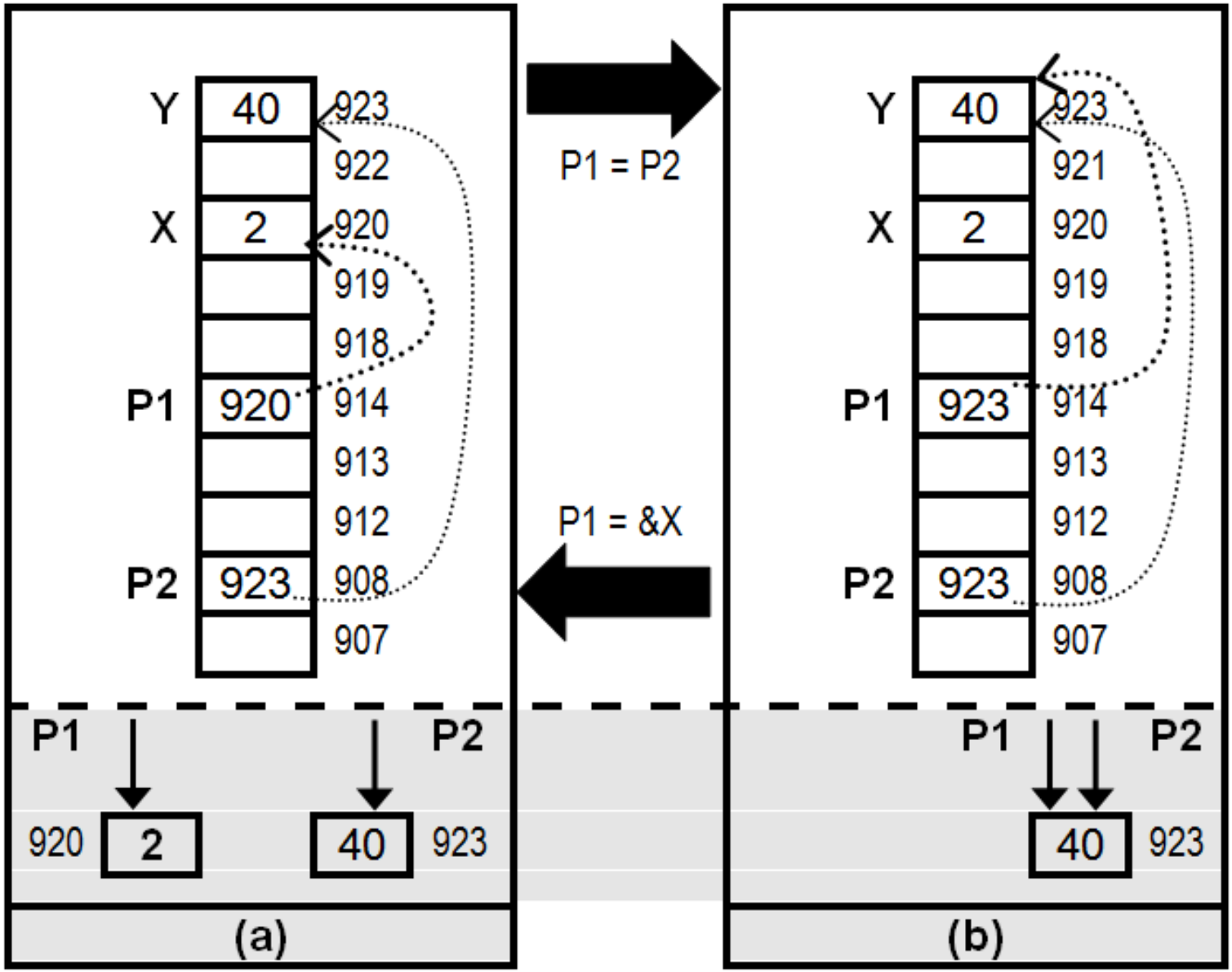
Atribuindo o Conteudo Apontado



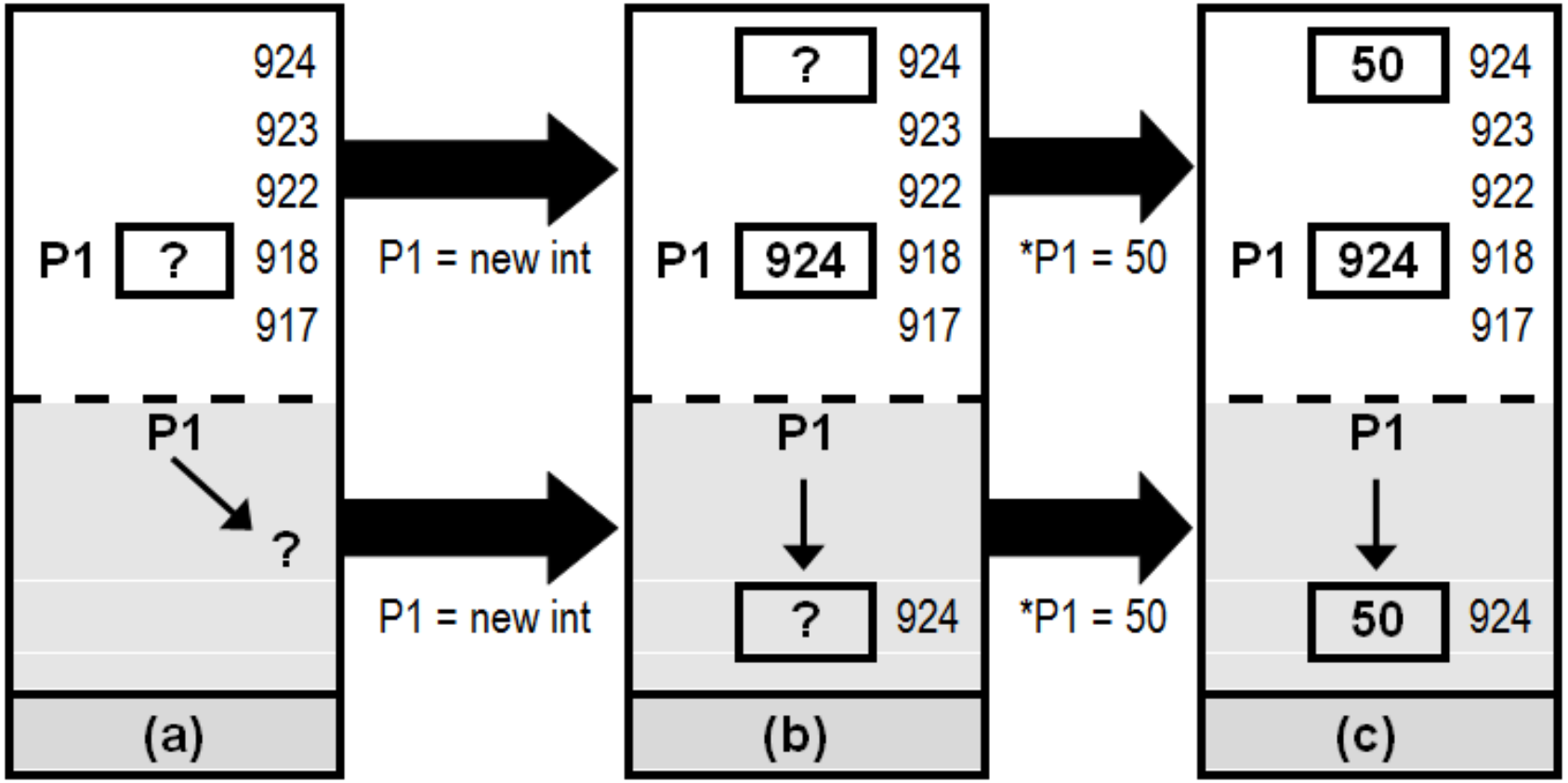
Atribuindo o Conteudo Apontado



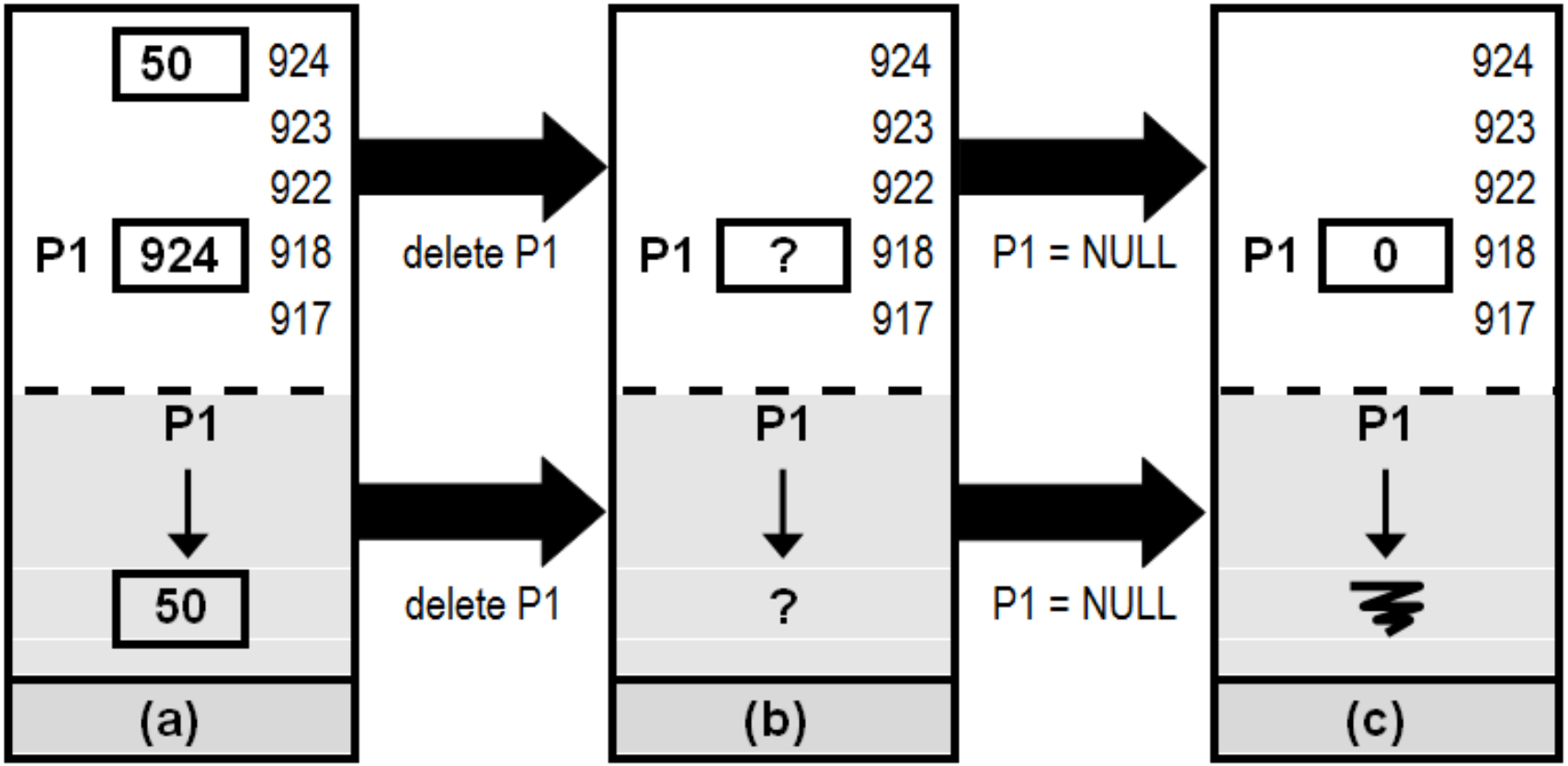
Movendo Ponteiros



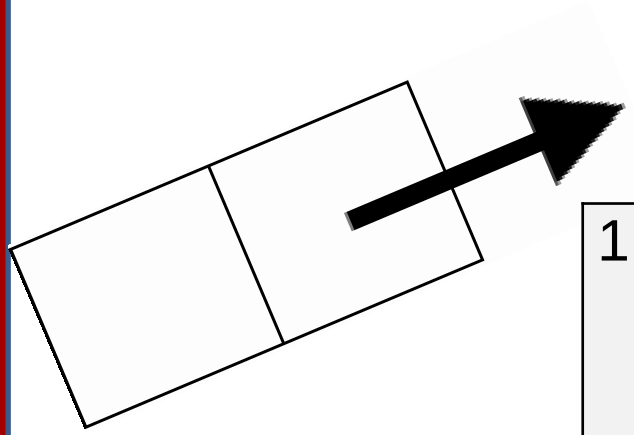
Alocando Memória Dinamicamente



Desalocando Memória Dinamicamente



Lista Encadeada Alocada Dinamicamente

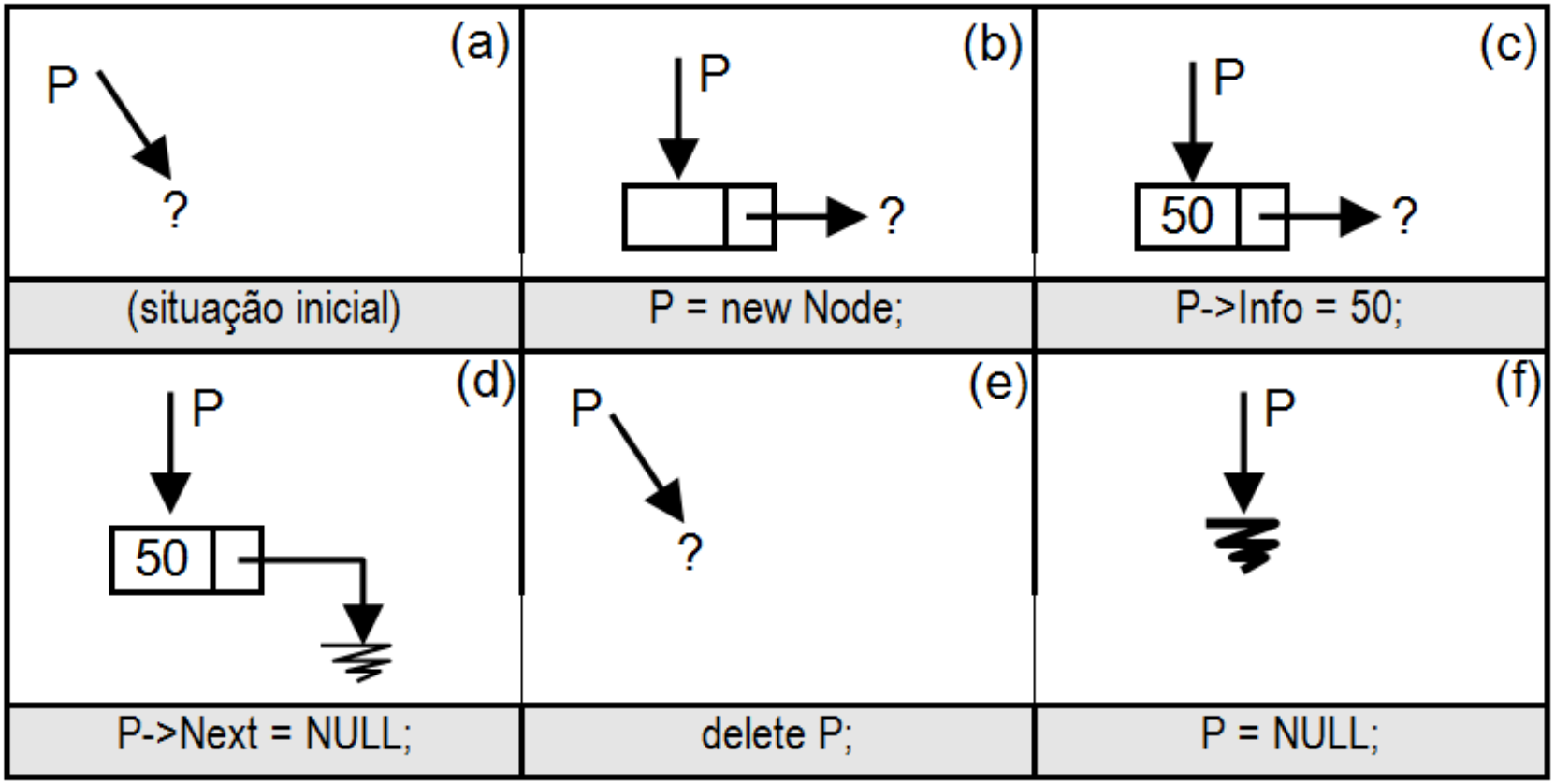


Comandos em C++

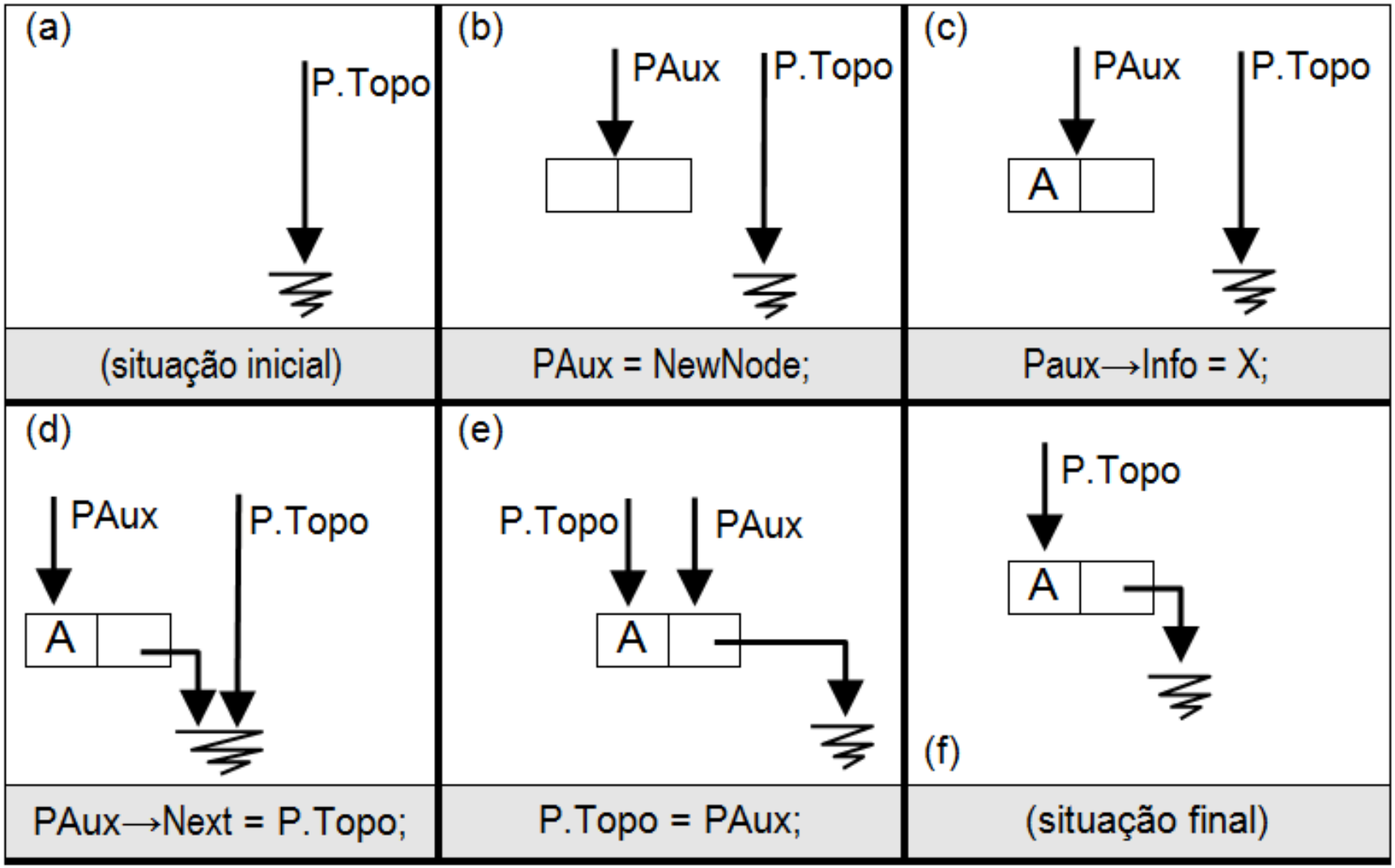
1	struct Node {
	char Info;
	struct Node *Next;
	};

2	typedef struct Node *NodePtr;
3	NodePtr P; PAux;
4	Int X;
5	P = new Node;
6	P->Info = 50;
7	P->Next = NULL;
8	X = P->Info;
9	PAux = P->Next;
10	Delete P;
11	P = NULL;

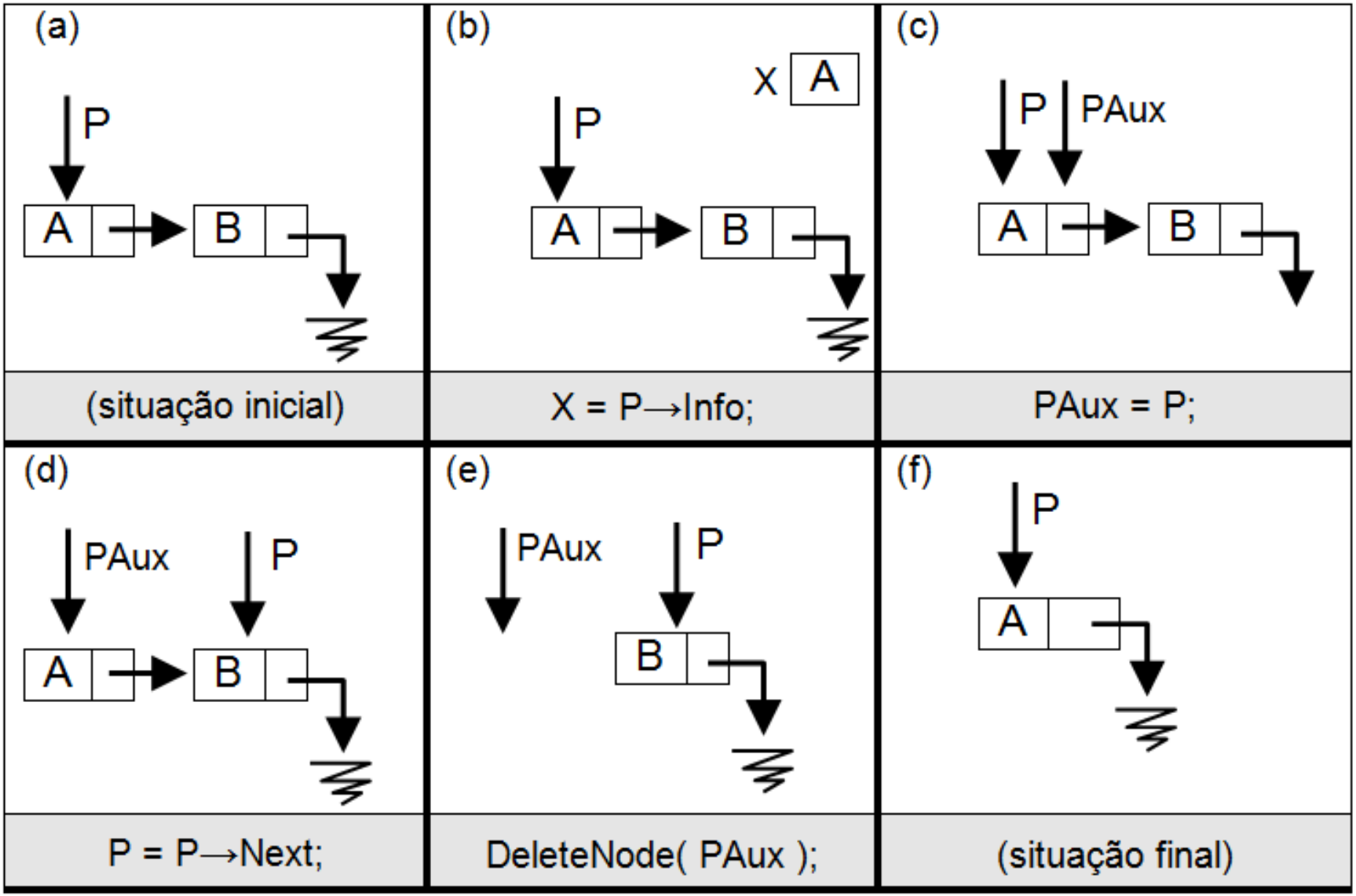
Lista Encadeada Alocada Dinamicamente: Comandos em C++



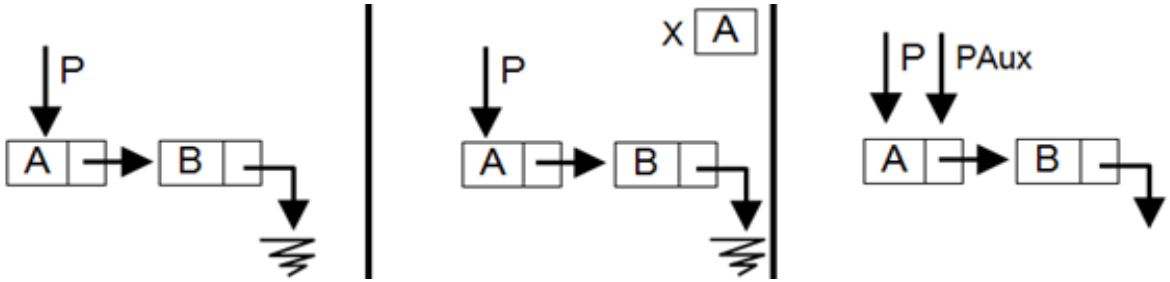
Exercício 5.1 Revisar Comandos da Operação Empilha



Exercício 5.2 Revisar Comandos da Operação Desempilha



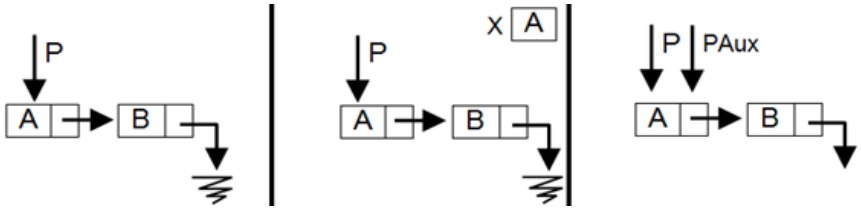
Dica Importante: Desenhe!



Ao elaborar e testar algoritmos sobre Listas Encadeadas, desenhe passo a passo!

A representação visual simplifica a compreensão, e evita erros.

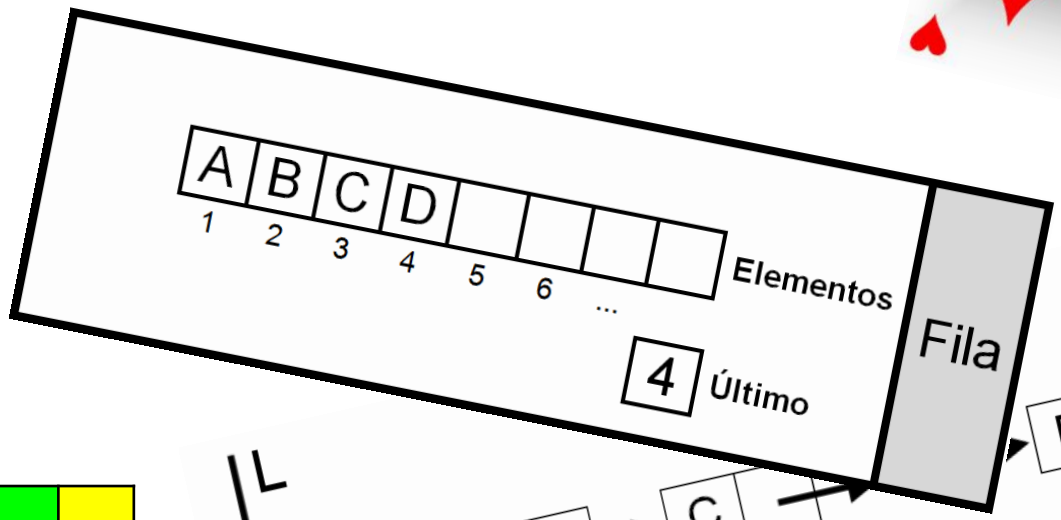
Exercícios

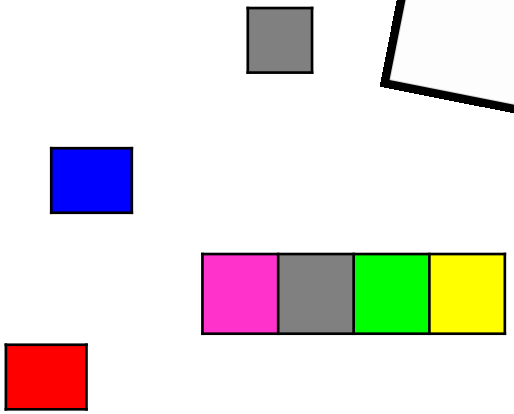


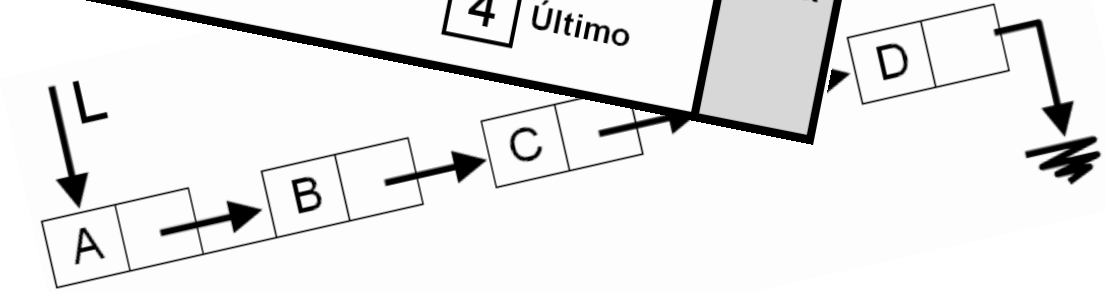
Exercício 5.3 Implemente uma Pilha com Alocação Encadeada e Dinâmica de Memória, em C++

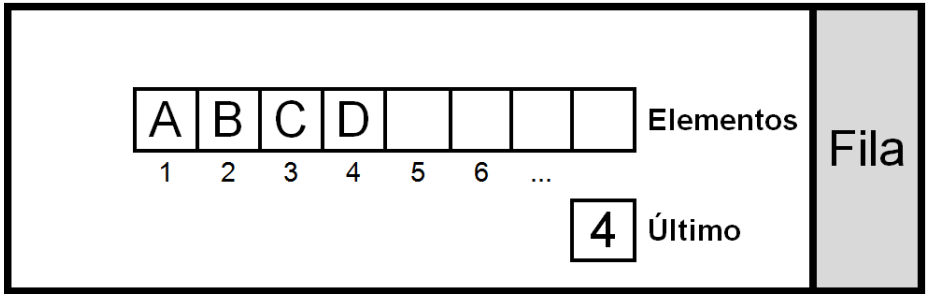
Exercício 5.4 Implemente uma Fila com Alocação Encadeada e Dinâmica de Memória, em C++

Alocação Sequencial e Estática ou Encadeada e Dinâmica?



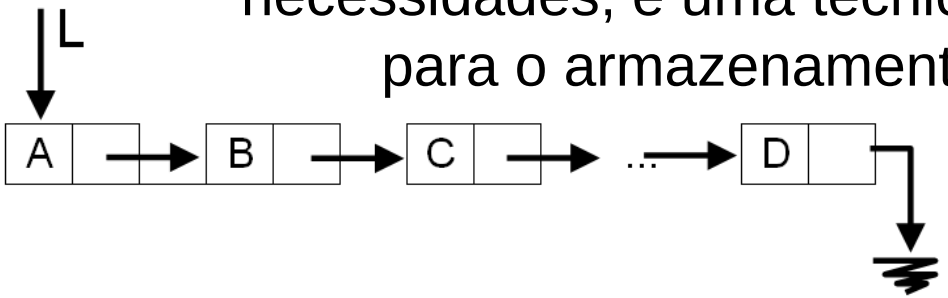






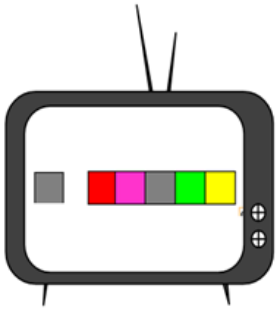
A Alocação **Sequencial** **e** **Estática** é uma técnica simples, e adequada a situações em que a quantidade de elementos que poderão entrar no conjunto é previsível, com pequena margem de variação.

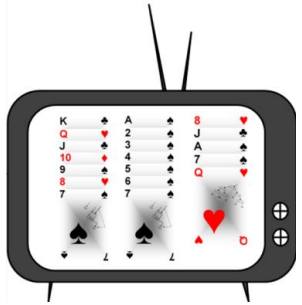
A Alocação **Encadeada** **e** **Dinâmica** é flexível com relação à quantidade de elementos, e pode ser facilmente adaptada para modelar diferentes necessidades; é uma técnica poderosa, e muito utilizada para o armazenamento temporário de conjuntos de elementos.

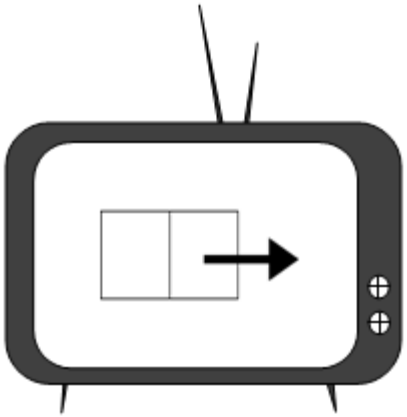


Exercícios de Fixação

Exercício 5.16 Avanço de Projeto: Avaliar a Portabilidade das Soluções com Pilha e Fila de Seus Jogos.







Exercício 5.13 Implemente uma Classe Node em C++.

Exercícios 5.5 e 5.6 Avanço de Projeto

Qual **combinação de técnicas** parece ser mais adequada às características dos jogos que você está desenvolvendo no momento: Alocação **Sequencial e Estática** ou Alocação **Encadeada e Dinâmica**?