

ESTRUTURA DE DADOS I

Ramon Lummertz

ESTRUTURA DE DADOS

Hoje em dia, a grande maioria das pessoas utilizam a agenda do celular para armazenar seus contatos. As tarefas de uma agenda de contatos são basicamente duas

Definir como as informações dos contatos serão armazenadas. Uma informação armazenada em algum lugar (pedaço de papel, livro, computador, etc) é um dado.

Disponibilizar operações para criar, recuperar, ordenar, atualizar e remover contatos. Além de operações para informar o estado da agenda (a quantidade de contatos existentes na agenda ou a quantidade de espaço disponível para novos contatos).



ESTRUTURA DE DADOS

Cada celular pode implementar a sua agenda de contatos de uma forma totalmente diferente um do outro, na tentativa de obter mais performance, ser mais confiável ou gastar menos memória. Porém o conjunto básico de operações oferecidas pelas agendas é sempre o mesmo. Isso facilita a vida do usuário pois se ele tiver que trocar de celular não vai ter que aprender novamente como usar uma agenda de contatos.

Uma agenda de celular pode ser vista como uma estrutura de dados. Uma estrutura de dados mantém os dados organizados seguindo alguma lógica e disponibiliza operações para o usuário manipular os dados.

É importante, quando programar, não misturar dado e estrutura de dados em uma coisa só. Um dado é uma informação armazenada e estrutura de dados é quem administra os dados. O ideal é que a estrutura de dados seja o mais independente possível dos dados que ela vai armazenar. Desta forma pode-se aproveitar a mesma estrutura de dados para diversos tipos de dados. Por exemplo, é melhor ter uma agenda genérica do que uma agenda de telefones. Uma agenda genérica pode ser utilizada para guardar telefones, ou emails, ou até mesmo guardar uma outra estrutura dentro dela: contatos, que seriam compostos por nome, telefone e email.

ESTRUTURA DE DADOS

As estruturas de dados, na maioria dos casos, baseiam-se nos tipos de armazenamento vistos dia a dia, ou seja, nada mais são do que a transformação de uma forma de armazenamento já conhecida e utilizada no mundo real adaptada para o mundo computacional. Por isso, cada tipo de estrutura de dados possui vantagens e desvantagens e cada uma delas tem sua área de atuação (massa de dados) otimizada.

ESTRUTURA DE DADOS

Os dados manipulados por um algoritmo podem possuir natureza distinta, isto é, podem ser números, letras, frases etc. Dependendo da natureza de um dado, algumas operações podem ou não fazer sentido quando aplicadas a eles.

Por exemplo, não faz sentido falar em somar duas letras - algumas linguagens de programação permitem que ocorra a soma dos valores ASCII correspondentes de cada letra.

ESTRUTURA DE DADOS

Para poder distinguir dados de naturezas distintas e saber quais operações podem ser realizadas com eles, os algoritmos lidam com o conceito de tipo de dados. O tipo de um dado define o conjunto de valores que uma variável pode assumir, bem como o conjunto de todas as operações que podem atuar sobre qualquer valor daquela variável.

Por exemplo, uma variável do tipo inteiro pode assumir o conjunto de todos os números e de todas as operações que podem ser aplicadas a estes números.

ESTRUTURA DE DADOS

Os tipos de dados manipulados por um algoritmo podem ser classificados em dois grupos: atômicos e complexos ou compostos. Os tipos atômicos são aqueles cujos elementos do conjunto de valores são indivisíveis, por exemplo: o tipo inteiro, real, caractere e lógico.

Por outro lado, os tipos complexos são aqueles cujos elementos do conjunto de valores podem ser decompostos em partes mais simples. Se um tipo de dado pode ser decomposto, então o tipo de dado é dito estruturado, e a organização de cada componente e as relações entre eles constituem a disciplina de Estrutura de Dados.

DADOS HOMOGÊNEOS

Uma estrutura de dados, que utiliza somente um tipo de dado, em sua definição é conhecida como *dados homogêneos*. Variáveis compostas homogêneas correspondem a posições de memória, identificadas por um mesmo nome, individualizado por índices e cujo conteúdo é composto do mesmo tipo. Sendo os vetores (também conhecidos como estruturas de dados unidimensionais) e as matrizes (estruturas de dados bidimensionais) os representantes dos dados homogêneos.

ESTRUTURAS UNIDIMENSIONAIS

O vetor é uma estrutura de dados linear que necessita de somente um índice para que seus elementos sejam endereçados. É utilizado para armazenar uma lista de valores do mesmo tipo, ou seja, o tipo vetor permite armazenar mais de um valor em uma mesma variável. Um dado vetor é definido como tendo um número fixo de células idênticas (seu conteúdo é dividido em posições). Cada célula armazena um e somente um dos valores de dados do vetor. Cada uma das células de um vetor possui seu próprio endereço, ou índice, através do qual pode ser referenciada. Nessa estrutura todos os elementos são do mesmo tipo, e cada um pode receber um valor diferente [3, 21, 4].

ESTRUTURAS UNIDIMENSIONAIS

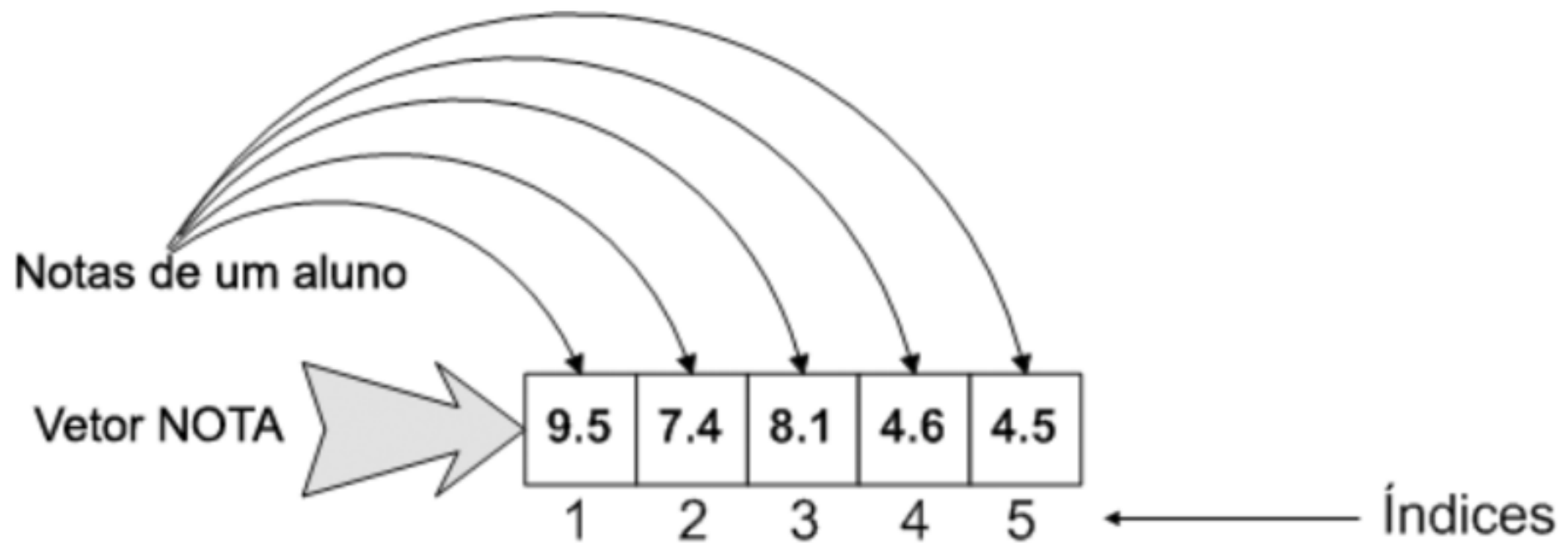


Figura 1.1: Exemplo de Vetor

CARACTERÍSTICAS DE VETORES

- Alocação estática (deve-se conhecer as dimensões da estrutura no momento da declaração)
- Estrutura homogênea
- Alocação seqüencial (bytes contíguos)
- Inserção/Exclusão
 - Realocação dos elementos
 - Posição de memória não liberada

CARACTERÍSTICAS DE VETORES

Programa 1.1: Declaração de vetor em C

```
1  int i[3];  
    i[0]=21;  
    i[1]=22;  
    i[2]=24;  
  
6  char c[4];  
    c[0]='a';  
    c[1]='b';  
    c[2]='c';  
    c[3]='d';
```

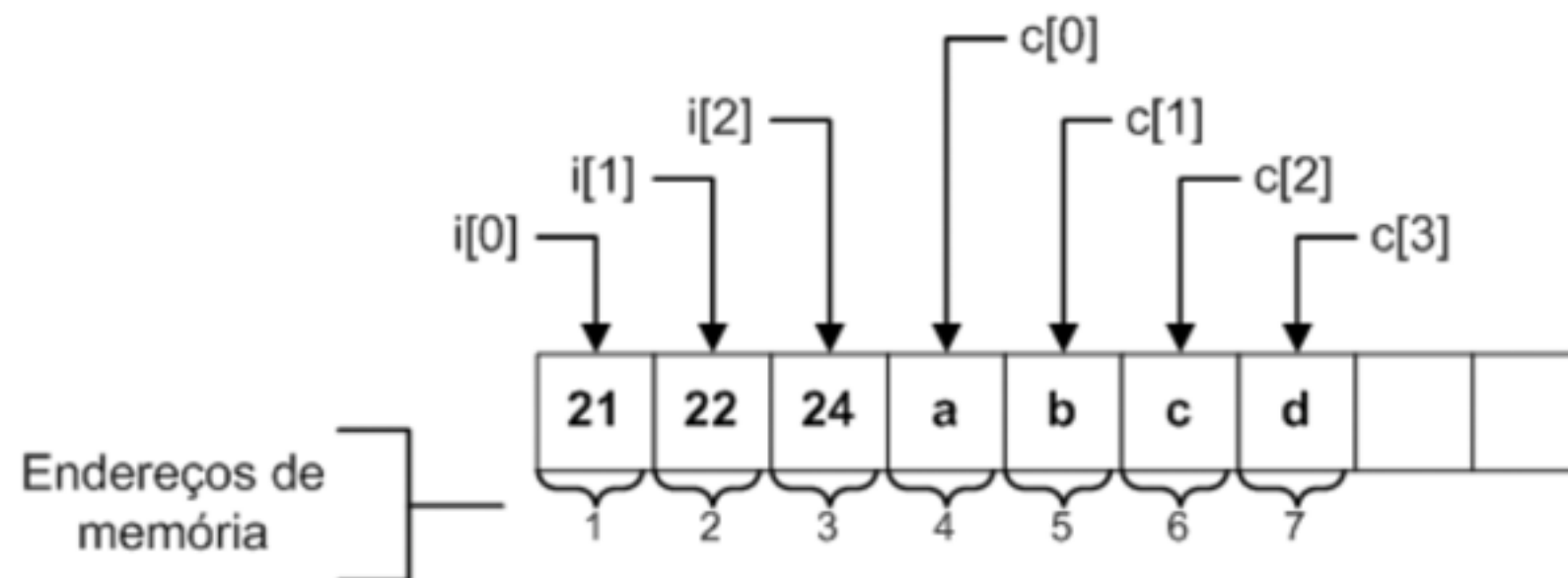


Figura 1.2: Representação de um vetor na memória

MATRIZES

Uma matriz é um arranjo bidimensional ou multidimensional de alocação estática e seqüencial. A matriz é uma estrutura de dados que necessita de um índice para referenciar a linha e outro para referenciar a coluna para que seus elementos sejam endereçados. Da mesma forma que um vetor, uma matriz é definida com um tamanho fixo, todos os elementos são do mesmo tipo, cada célula contém somente um valor e os tamanhos dos valores são os mesmos. (em C, um `char` ocupa 1 *byte* e um `int` 4 *bytes*)

Os elementos ocupam posições contíguas na memória. A alocação dos elementos da matriz na memória pode ser feita colocando os elementos linha-por-linha ou coluna-por-coluna.

EXEMPLO EM JAVA

```
public class Main {  
    public static void main(String[] args) {  
        // write your code here  
  
        int vetor[] = new int [5]; // vetor de 5 elementos  
        System.out.println(vetor.length); // mostra o tamanho do vetor  
        int matriz[][] = new int [5][4]; // matrix de 5x5 elementos  
        System.out.println(matriz.length); // numero de linhas  
        System.out.println(matriz[0].length); // numero de colunas  
    }  
}
```

MATRIZES

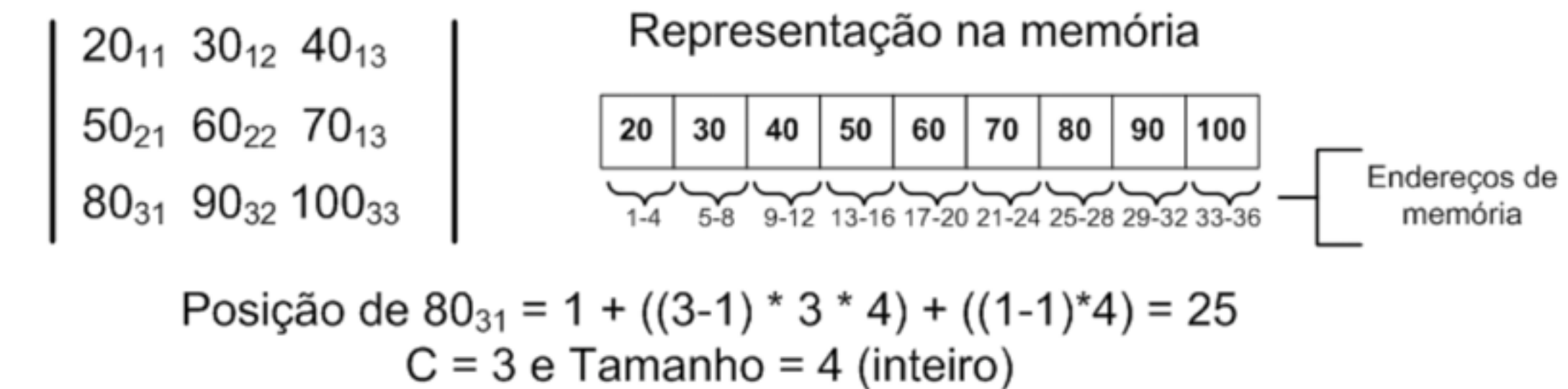
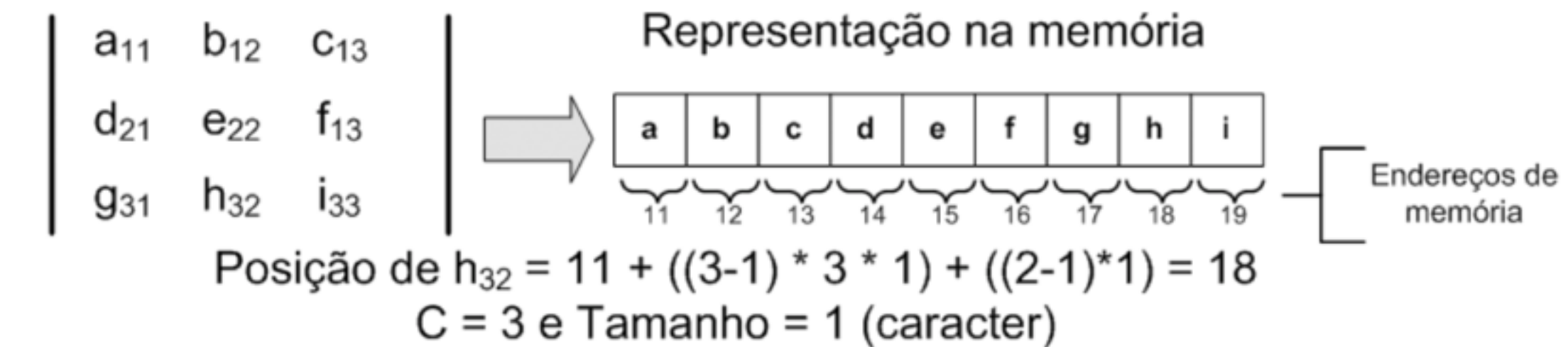


Figura 1.4: Cálculo de posição na memória

ATIVIDADES

1. Dadas as temperaturas que foram registradas diariamente durante uma semana, deseja-se determinar em quantos dias dessa semana a temperatura esteve acima da média. A solução para esse problema envolve os seguintes passos:
 - a) obter os valores das temperaturas;
 - b) calcular a média desses valores;
 - c) verificar quantos deles são maiores que a média.

1. Dados os vetores $A = [15, 44, 23, 1, 0, 18, 17, 37, 35, 54]$ e $B = [32, 115, 48, 55, 51, 0, -48, 85, 15, 99]$, crie algoritmos e programa para gerar uma matriz C a partir da:
 - a) multiplicação dos elementos de A por B ;
 - b) adição dos elementos de A e B ;
 - c) subtração dos elementos de A de B ;
 - d) $A \cup B$.

ATIVIDADES

20. Faça um programa que leia um vetor com cinco posições para números reais e, depois, um código inteiro. Se o código for zero, finalize o programa; se for 1, mostre o vetor na ordem direta; se for 2, mostre o vetor na ordem inversa.

16. Faça um programa que preencha um vetor com dez números inteiros e um segundo vetor com cinco números inteiros, calcule e mostre dois vetores resultantes. O primeiro vetor resultante será composto pelos números pares gerados pelo elemento do primeiro vetor somado a todos os elementos do segundo vetor; o segundo será composto pelos números ímpares gerados pelo elemento do primeiro vetor somado a todos os elementos do segundo vetor.

Primeiro vetor	4	7	5	8	2	15	9	6	10	11
	1	2	3	4	5	6	7	8	9	10

Segundo vetor	3	4	5	8	2
	1	2	3	4	5


Primeiro vetor resultante

8 + 3 + 4 + 5 + 8 + 2				
26	30	24	...	
4 + 3 + 4 + 5 + 8 + 2				

Segundo vetor resultante

7 não possui divisores				
0	1	2	...	
5 é divisível apenas por 5		15 é divisível por 3 e por 5		

ATIVIDADES

 13. Faça um programa que preencha um vetor com dez números inteiros, calcule e mostre os números superiores a cinquenta e suas respectivas posições. O programa deverá mostrar mensagem se não existir nenhum número nessa condição.

3. Uma empresa deseja saber a média de idade dos frequentadores de sua praça de alimentação. Para isso, você deve construir uma matriz que seja capaz de armazenar a idade de cem pessoas.
 - a) Seria possível armazenar também os nomes dos participantes da pesquisa? Como resolver esse problema?
 - b) Identifique a maior idade da matriz.
 - c) Calcule a média das idades da matriz.
 - d) Identifique a menor idade da matriz.