Plano de aula

Clube de Programação UTFPR

October 11, 2025

Sumário

1 Arrays N-Dimensionais (estimativa: 20 min)

1.1 O que são e para que servem?

Um array (também chamado de arranjo ou vetor) é uma estrutura de dados que armazena uma coleção ordenada de elementos, onde todos os itens são geralmente do mesmo tipo (como int ou string).

Diferente de uma variável comum que guarda apenas um valor, um array pode conter múltiplos valores em uma única variável. Cada elemento no array é acessado através de um índice numérico, que indica sua posição na sequência (começando em 0 na maioria das linguagens).

Uma característica fundamental é que seus elementos são armazenados em um bloco de memória contígua, o que permite um acesso quase instantâneo a qualquer elemento através de seu índice.

1.2 Declaração e Acesso em C++ (`std::vector`)

```
1     #include vector  // usamos #include bits/stdc++.h //
2
3     int tamanho = 10;
4     vector<int>nome(tamanho);
5     //comentar sobre sem tamanho especificado
6
```

1.3 Percorrendo Vetores

```
1     for(int i=0;i<tamanho;i++){
2     cout << nome[i] << " ";
3     }
4
```

1.4 Funções disponíveis com o std::vector

```
1     vector<int>nome;
2     nome.push_back(valor);
3     nome.size();
4     nome.empty();
5     nome.pop_back();
6     nome.clear();
7     nome.resize(tamanho);
8
9     comento sobre funcoes que envolvem iterador?
10
```

1.5 percorrer matriz

```
1     vector<vector<int>>matriz(tamanho,vector<int>(tamanho));
2     for(int i=0;i<matriz.size();i++){
3         for(int j=0;j<matriz[i].size();j++){
```

```
4           cout << matriz[i][j]<< " ";
5           }
6           cout<< endl;
7       }
8
```

1.6 Aplicações Comuns

- **Modelar problemas:** Grids, conjunto de valores..
- **Base para muitas Estruturas de dados**
- **Programação Dinâmica:** Tabelas de memoização.
- **Representação de Grafos:** Matrizes de adjacência.

1.7 Problema de motivação (Diferenca de diagonais)

```
1       #inlude <bits/stdc++.h>
2       using namespace std;
3
4       cin>>n;
5       vector<vector<int>> matriz(n, vector<int>(n));
6
7       for (int i = 0; i < n; i++) {
8           for (int j = 0; j < n; j++) {
9               cin >> matriz[i][j];
10          }
11      }
12
13      long long sum1 = 0;
14      long long sum2 = 0;
15
16      for (int i = 0; i < n; i++) {
17          sum1 += matriz[i][i];
18          sum2 += matriz[i][n - 1 - i];
19      }
20
21      long long diferenca = abs(sum1 - sum2);
22      cout << diferenca << endl;
23  }
24
```

2 Strings (estimativa: 15 min)

2.1 O que são?

Como dito anteriormente, arrays podem ser de vários tipos, como char.

Para formar uma palavra, precisamos de muitos caracteres reunidos...

Uma string é um tipo de dado em programação que representa uma sequência de caracteres, como letras, números e símbolos. É utilizada para armazenar texto em um programa e é fundamental para manipular dados textuais, como nomes, frases, endereços e senhas.
Podem ser concatenadas!!!

2.2 Manipulação Básica em C++ (`std::string`)

```
1     #include <bits/stdc++.h>
2     using namespace std;
3
4     int main(){
5         int tamanho = 20;
6         string s1 = "Clube";
7         string s2 = " de Prog";
8         cout << s1 + s2 << endl;
9         cout << s1[0] << endl;
10        cout << s1.size() << endl;
11    }
12
```

2.3 Problema de motivacao: informar se o num /3

```
1     #include <bits/stdc++.h>
2     using namespace std;
3
4     int main(){
5         int n,cont=0;
6         string s;
7         cin>>n;
8         cin>>s;
9
10        for(int i=0;i<n;i++){
11            cont+=s[i]-48;
12        }
13
14        if(cont%3==0){
15            cout<<"YES\n"<<endl;
16        }
17        else{
18            cout<<"NO\n"<<endl;
19        }
20    }
21
```

3 Contagem de Frequências (estimativa: 15min)

3.1 Técnica 1: Vetor de Frequência

Esta técnica é ideal sob duas condições principais:

- Quando os elementos que você quer contar são **números inteiros** ou **caracteres**.
- Quando o intervalo de valores é **pequeno e conhecido**. Por exemplo:
  - Letras minúsculas do alfabeto ('a' a 'z' → 26 valores).
  - Caracteres da tabela ASCII (→ 128 ou 256 valores).

Obs: O vetor de frequências é inflexivel.. Se os valores forem muito grandes (ex: 10^9) ou não forem numéricos (ex: strings), a melhor opção é usar um `std::map` (que seria a outra técnica de contagem de frequências).

3.2 Problema de motivaçao: informar a quantidade de letras do alfabeto faltante

```
1         #include <bits/stdc++.h>
2
3     using namespace std;
4
5     int main() {
6         string s;
7         vector<bool> charac(26, false);
8         cin >> s;
9
10        for (char c : s) {
11            s[c - 'a'] = true;
12        }
13        int cont=0;
14
15        for(int i=0;i<charac.size();i++){
16            if(!charac[i]){
17                cont++;
18            }
19        }
20        cout << cont << endl;
21
22        return 0;
23    }
24
25
```