

Avaliação Prática 01

Estrutura de Dados - Prof. Hamilton José Brumatto

Problema A: Retas Paralelas

Arquivo: paralela.[h,hpp,py];

O Problema:

As retas e outros elementos geométricos são bons exemplos de aplicação de estrutura de dados. Uma reta é definida por uma equação: $ax + by + cz = d$, ou seja, os coeficientes a , b , c e d definem unicamente uma reta. Vamos nos ater a uma reta no plano, neste caso z deixa de ser uma variável, e podemos nos restringir $ax + by = c$, neste caso uma estrutura contendo a , b e c definem uma reta no plano. Este tema é explorado em geometria computacional, mas vamos ficar somente com isto.

Neste problema queremos uma única informação, se duas retas são paralelas. Isto é fácil verificar:

- Ambas retas possuem $a = 0$ ou $b = 0$, então são paralelas.
- Se $a_1/b_1 = a_2/b_2$, então são paralelas.

Um problema na computação com números em ponto flutuante é que a comparação de igualdade nem sempre resulta de forma correta, é interessante tomar um valor ϵ muito pequeno ($< 10^{-100}$) e verificar se $a_1/b_1 - a_2/b_2 < \epsilon$ como teste para verificar se as retas são paralelas.

Vamos criar uma biblioteca com o tipo “reta” para que seja usada em códigos. Crie uma estrutura / classe **reta** para representar uma reta e as funções / métodos que irão criar uma reta e verificar se duas retas são paralelas. Implemente, também, estes métodos/funções.

Para a linguagem C temos os protótipos:

```
typedef struct reta { → declaração do tipo
    ...
} reta;
...
void criar(reta *r, double a, double b, double c); → cria uma reta com  $a, b, c$ .
int paralela(reta r, reta s) → retorna 0 se duas retas não são paralelas ou qualquer outro valor, caso contrário.
```

Para a linguagem C++ temos os protótipos:

```
class reta { → declaração da classe
    ...
public:
    → permite que manipule-se a classe a partir da “main”
    reta(double a, double b, double c); → construtor cria objeto reta com  $a, b, c$ .
    bool paralela(reta s); → verifica se a reta  $s$  é paralela à reta deste objeto.
};
```

Para a linguagem Python temos os protótipos:

```
class reta: → declaração da classe
    ...
    def __init__(self, a, b, c): → construtor cria objeto reta com  $a, b, c$ .
    ...
    def paralela(self, s) → verifica se a reta  $s$  é paralela à reta deste objeto.
    ...
```