

Expressões aritméticas

Uma estrutura de dados que a gente costuma utilizar para realizar raciocínios matemáticos são as expressões aritméticas

(3 + ((2 × ((15 + 1) / (13 - 9))) + 5))

Daí, se alguém pede para você calcular essa expressão, é possível que você comece com a operação

15 + 1

Porque?

Ora, porque a gente aprendeu que deve realizar as operações dentro dos parênteses primeiro.

E essa operação se encontra dentro de um par de parênteses mais interno — (i.e., não existem parênteses dentro desse par de parênteses)

Logo, a gente pode começar por ali.

Certo.

Mas agora a gente pergunta

- Como é que um computador pode calcular o valor dessa expressão ?

Bom, a primeira coisa é que a expressão vai estar armazenada em uma lista

V

(	3	+	(	(	2	*	(	(	15	+	1	)	÷	(	13	-	9	)	)	)	+	5	)	)
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---

Mas, e agora?

Como é que a gente localiza um par de parênteses mais interno?

Bom, aqui uma pequena esperteza resolve o nosso problema.

Quer dizer, basta localizar o primeiro fecha-parênteses

V

(	3	+	(	(	2	*	(	(	15	+	1	)	÷	(	13	-	9	)	)	)	+	5	)	)
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---

↑

Daí, andando para trás a gente encontra o abre-parênteses correspondente.

E agora, a gente pode fazer essa conta

V

(	3	+	(	(	2	*	(	(	15	+	1	)	÷	(	13	-	9	)	)	)	+	5	)	)
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---

↑ 16 ↑

Mas, e agora?

Quer dizer, onde é que a gente armazena o 16?

Porque o 16 é apenas um número, mas existem 5 posições que vão ficar vazias.

Bom, aqui é a hora para mais uma esperteza.

Quer dizer, a ideia é ir copiando a expressão para uma lista auxiliar, à medida que a gente procura o fecha-parênteses

V

(	3	+	(	(	2	*	(	(	15	+	1	)	÷	(	13	-	9	)	)	)	+	5	)	)
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---

↑

P

(	3	+	(	(	2	*	(	(	15	+	1	)												
---	---	---	---	---	---	---	---	---	----	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--

Daí, ao invés de fazer essa conta na lista original, a gente faz a conta na lista auxiliar.

O que deixa a coisa assim

V

(	3	+	(	(	2	*	(	(	15	+	1	)	÷	(	13	-	9	)	)	)	+	5	)	)
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---

↑

P

(	3	+	(	(	2	*	(	16															
---	---	---	---	---	---	---	---	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Agora, é só continuar a leitura até encontrar o próximo fecha-parênteses

V

(	3	+	(	(	2	*	(	(	15	+	1	)	÷	(	13	-	9	)	)	)	+	5	)	)
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---

↑

P

(	3	+	(	(	2	*	(	(	16	÷	(	13	-	9	)								
---	---	---	---	---	---	---	---	---	----	---	---	----	---	---	---	--	--	--	--	--	--	--	--

E daí, a gente continua fazendo a mesma coisa até o fim.

Em resumo, a ideia é a seguinte

- a gente vai lendo os elementos de V e copiando para P, até encontrar um fecha-parênteses
- daí, a gente faz a conta dentro do parênteses em P
- e continua fazendo isso até o fim

E o algoritmo fica assim

```
V = { (, 3, +, (, 2, *, (, ...}
P = {}

i = 1 // coloca o dedo no início
j = 0 // não há nada em P

Enquanto ( i <= N )

    j++; P[j] = V[i] // copia termo para P

    Se ( V[i] == fecha-parênteses )

        j--; N2 = P[j]
        j--; op = P[j]
        j--; N1 = P[j]

        res = Calcula ( N1, op, N2 )

        j--; P[j] = res

    i++

Imprime ( "o resultado é:", P[1] )
```

Legal, não é?