

CALCULADORA

ADIÇÃO E
MULTIPLICAÇÃO



FURG
UNIVERSIDADE FEDERAL
DO RIO GRANDE

ALGORITMOS E
ESTRUTURA E DADOS

ANDRESSA LEAL CONSTANTINO - 149462

RELATÓRIO

DESCRIÇÃO DO CÓDIGO

O algoritmo em calculadora.py contem uma função de soma que armazena o numero como string. Primeiro, ocorre uma verificação se os números são do mesmo tamanho, logo depois são criadas variáveis auxiliares e então começa o loop que converte dígito por dígito para numero inteiro para realizar as operações necessárias, o mesmo ocorre na função de multiplicação, no final, o resultado final é invertido. Na multiplicação todos os resultados das pequenas multiplicações são armazenados em um vetor para serem somados posteriormente com a função de soma

PSEUDO-CÓDIGO (CALCULADORA.PY)

função de soma (num1, num2):

se o primeiro numero conter mais digitos que o segundo:

diferença = tamanho primeiro numero - tamanho segundo

numero

loop para salvar string com a quantidade de zeros da

diferença

segundo numero = zeros + segundo numero

se o segundo numero conter mais digitos que o primeiro:

diferença = tamanho segundo numero - tamanho primeiro

numero

loop para salvar string com a quantidade de zeros da

diferença

segundo numero = zeros + segundo numero

tamanho = tamanho primeiro numero

resto = nada

resultado = string vazia

percorrendo o numero de tras pra frente:

somar = dígito i do primeiro numero + dígito i do segundo +

resto

se o resultado conter mais de 1 dígito:

armazena o segundo dígito no resultado e guarda o

primeiro como resto

PSEUDO-CÓDIGO (CALCULADORA.PY)

caso contrario:

armazena o resultado como resultado

se depois do loop tiver sobrado resto:

resultado = resultado + resto

inverter resultado (pois os numeros foram adicionados da esquerda pra direita)

retorna resultado

função de multiplicação (num1, num2):

se o primeiro numero conter mais digitos que o segundo:

diferença = tamanho primeiro numero - tamanho segundo

numero

loop para salvar string com a quantidade de zeros da diferença

segundo numero = zeros + segundo numero

se o segundo numero conter mais digitos que o primeiro:

diferença = tamanho segundo numero - tamanho primeiro

numero

loop para salvar string com a quantidade de zeros da diferença

segundo numero = zeros + segundo numero

tamanho = tamanho primeiro numero

resto = nada

resultado = string vazia

vetor para colocar numeros que devem ser somados depois da multiplicação

casasNum = variavel para contar o numero de casas que precisa deixar em branco à direita a cada multiplicação

loop percorrendo o tamanho do numero de tras pra frente:

resultado = string vazia

loop percorrendo o tamanho do numero de tras pra frente:

multiplicação = digito i do segundo numero * digito j do primeiro numero + resto

resto = nada

se o resultado conter mais de 1 digito:

armazena o segundo digito no resultado e guarda o primeiro como resto

caso contrario:

armazena o resultado como resultado

PSEUDO-CÓDIGO (CALCULADORA.PY)

se for o ultimo digito do numero e resto for diferente de zero:

adicionar resto ao resultado
resto = nada

inverter resultado
adicionar casasNum zeros ao final
adicionar resultado ao vetor de numeros para somar

casasNum += 1

loop percorrendo o vetor:

zeros = string vazia
diferença = tamanho(ultimo numero) - tamanho(numero i)
zeros = diferença * zeros
auxiliar = numero i
numero i = zeros + auxiliar

auxiliar = string vazia

loop percorrendo o vetor:

se for o primeiro numero do vetor:
auxiliar = numero i

senão:
resultado = somar(auxiliar, numero i)
auxiliar = resultado

variavel auxiliar = True
resultado = string vazia

loop percorrendo o resultado:

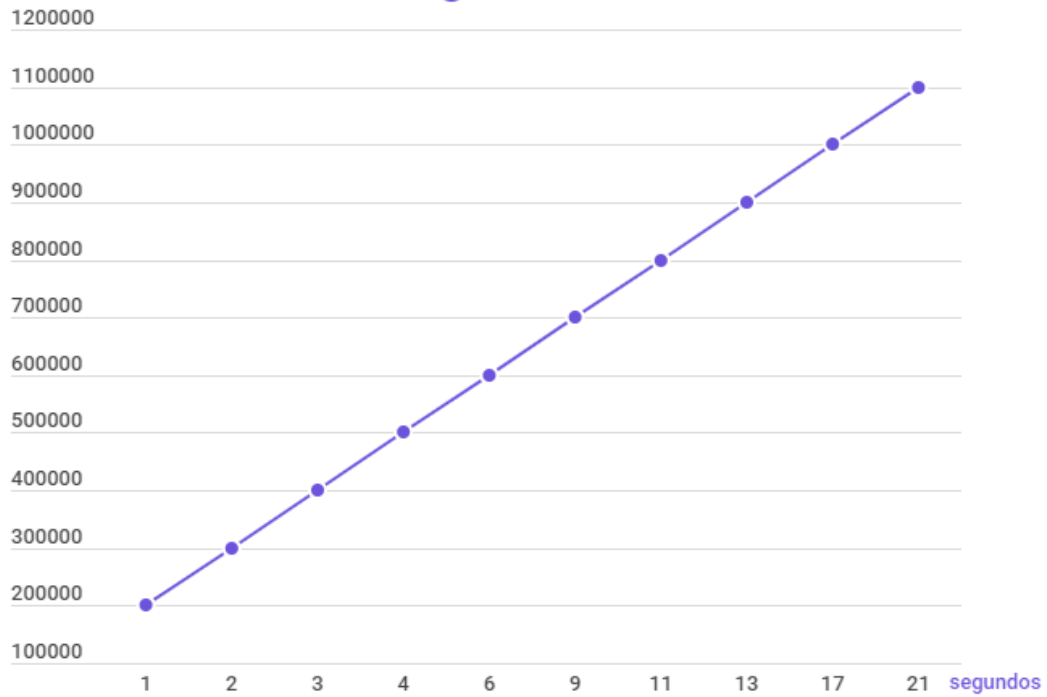
se digito i == 0 e variavel auxiliar == True:
faz nada
senão:
auxiliar = False
resultado += digito i

retorna resultado

GRÁFICOS

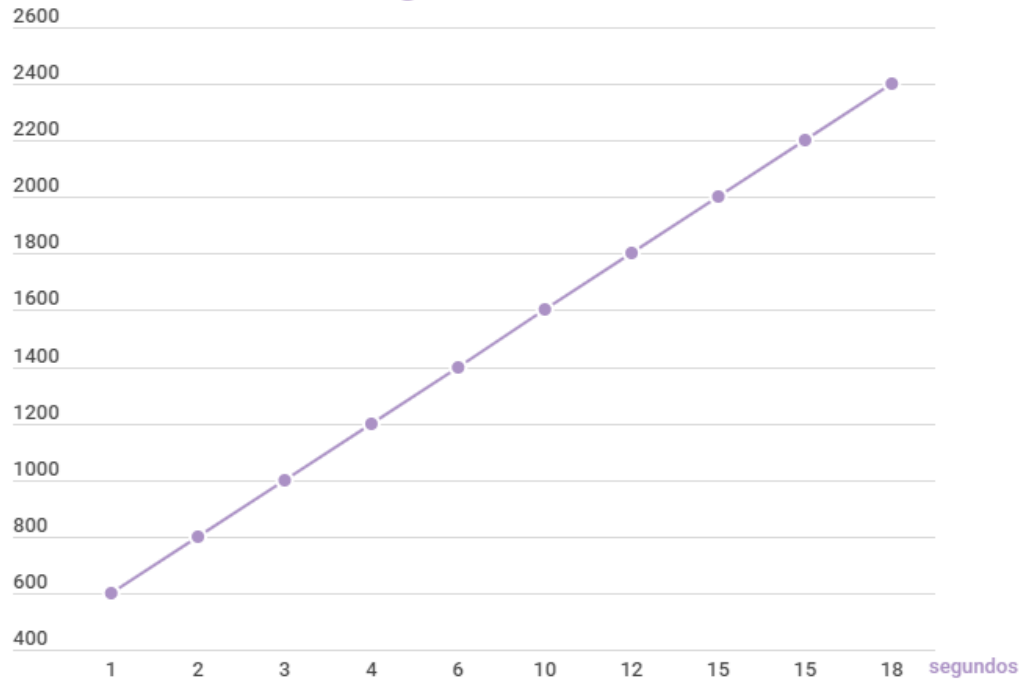
quantidade
de caracteres

● Soma



quantidade
de caracteres

● Multiplicação



Especificações detalhadas do computador escolhido

Aspire A315-42G

Processador: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10Ghz

RAM instalada: 8GB (utilizável até 5,94GB)

Tipo de sistema: Sistema operacional de 64 bits, processador baseado em x64

Windows:

Edição: Windows 10 Home Single Language

Versão: 21H1

ANÁLISE DOS RESULTADOS OBTIDOS E CONCLUSÕES FINAIS

Qual algoritmo apresentou melhor desempenho para entradas pequenas?

Para entradas grandes? O algoritmo de soma executa em menos tempo que o de multiplicação tanto entradas grandes quanto pequenas, então o algoritmo de soma apresentou o melhor desempenho em ambos os casos.

O crescimento do tempo de execução em relação ao tamanho de entrada na prática reflete o comportamento esperado? Sim, é esperado que o tempo de execução aumente enquanto o tamanho do input aumenta.

Como o uso de um processador/computador diferente afetaria o comportamento dos algoritmos? Pode variar no tempo de execução do código.

Qual função matemática melhor representaria a evolução do tempo de execução em função do tamanho de entrada para cada algoritmo?

Soma: $n/200000 + n-1/200000$ (onde n = numero de dígitos)

Multiplicação: $n/200 + [(n/2)/200] - 2$ (onde n = numero de dígitos)