

Projeto e Análise de Algoritmos

Terceiro Trabalho Prático (Primeira Versão)

Entrega Sugerida: 15/02/2025

2º Semestre 2024 - DC-UFSCar

1 Introdução

No terceiro Trabalho Prático (TP03) será solicitada a entrega de programas que solucionem os problemas apresentados nas próximas seções. Os programas entregues devem seguir os itens abaixo:

- Os programas devem ser feitos individualmente e plágio não será tolerado;
- Os programas devem ser entregues no beecrowd (<https://www.beecloud.com.br>) com código em linguagem C, contendo um cabeçalho com as informações do estudante (nome, curso, RA);
- Cada estudante deve se cadastrar no beecrowd (<https://www.beecloud.com.br>), acessar o beecrowd academic (<https://www.beecloud.com.br/judge/pt/disciplines>) e se matricular na disciplina “PAA 2024s2 Turmas A, B e C” usando Chave CyWtic8 e ID 013672;
- Na linguagem C, compile seus programas usando o compilador GCC com flags -Wall -pedantic -O2 -Wno-unused-result, pois warnings podem impedir o código de funcionar no beecrowd ainda que funcionem no seu computador.
- Importantel: Para todos os estudantes de computação (engenharia ou ciência), gostaria que o trabalho fosse feito em linguagem C, como está especificado no enunciado.
- Importante2: Eu vou liberar uma lista para submissão em Python para que os estudantes de outros cursos possam usar esta linguagem, já que eles não veem introdução à programação em C.

Notem que os problemas que formam este trabalho prático fazem parte da lista de exercícios “Programação Dinâmica” na página da disciplina no beecrowd.

2 Baú da Felicidade

Esse problema não é sobre o Silvio Santos. Ele é sobre o Henrico e o Frederico, que agora estão ricos. Acontece que eles foram para Praia Grande na baixada santista e, cavando um buraco na areia, acharam um baú do tesouro. Esse baú continha moedas de ouro, que valem mais que dinheiro, de pesos diferentes. Quanto mais pesada uma moeda, maior o seu valor.

Não sabendo muito bem o que fazer com esse ouro todo, eles decidiram primeiro dividir as moedas de forma justa e depois cada um decidia o que queria fazer com seu próprio dinheiro. Porém, eles não sabem muito bem como fazer isso. O que eles querem é que, dado o peso em gramas de cada moeda encontrada, você os ajude a encontrar qual é a menor diferença entre a quantidade de ouro para cada um deles após distribuir as moedas.

Dica: é possível reduzir este problema a um problema estudado nas aulas sobre Programação Dinâmica.

Entrada

A instância deve ser lida da entrada padrão. Cada instância contém vários testes, sendo que os dados de cada teste estão em uma única linha. O primeiro valor de cada linha indica o número N ($2 \leq N \leq 100$) de moedas encontradas naquele teste. Os próximos N números da linha correspondem aos pesos W_i ($1 \leq W_i \leq 500$) de cada moeda i , para $i = 1, \dots, N$. Todos os valores de entrada são inteiros.

O final da entrada é indicado por uma linha apenas com o número 0.

Saída

Para cada teste da instância, seu programa deve imprimir um único valor inteiro por linha na saída, indicando qual é a menor diferença possível entre os valores ganhos por Henrico e Frederico em cada teste.

Exemplos de entradas e saídas correspondentes

Entrada	Saída	Entrada	Saída
3 2 3 5	0	5 1 1 1 1 1	1
4 1 2 4 6	1	3 3 3 3	3
0		0	

3 Subconjuntos Divisíveis

O problema que queremos resolver é o de determinar o número de subconjuntos cuja soma é divisível por um certo número. Neste problema recebemos dois números inteiros não negativos N e D , e devemos dizer quantos subconjuntos de $[N] = \{1, \dots, N\}$ tem sua soma divisível por D . Por exemplo, seja $N = 3$ e $D = 4$. Os subconjuntos de $[N] = \{1, 2, 3\}$ são $\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}$ e $\{1, 2, 3\}$, cujas somas são, respectivamente, 0, 1, 2, 3, 3, 4, 5 e 6. Dessas, apenas 2 são divisíveis por $D = 4$.

Dica: para instâncias pequenas é possível usar algoritmos de enumeração de subconjuntos, vistos em AED1¹, mas para as grandes não acho que dê pra fugir de Programação Dinâmica :).

Curiosidade: o vídeo Olympiad level counting² do canal 3Blue1Brown fala mais sobre este problema, embora os métodos que ele apresenta não sejam necessários para nossa solução.

Entrada

A instância deve ser lida da entrada padrão. Cada instância contém vários testes, sendo que os dados de cada teste estão em uma única linha. Cada linha informa o tamanho N de um conjunto (em cujos subconjuntos estamos interessados) e o divisor D . A execução deve terminar quando lermos uma linha cujo primeiro valor é 0.

Saída

Para cada linha da entrada em que lemos os valores N e D , deve ser impresso na saída padrão o número de subconjuntos de $\{1, \dots, N\}$ cuja soma é divisível por D .

Exemplos de entradas e saídas correspondentes

Entrada	Saída	Entrada	Saída
1 5	1	4 2	8
3 4	2	3 7	1
0		7 3	44
		10 5	208
		0	

¹<http://www.aloc.ufscar.br/felice/ensino/2022s2aed1/aed1.php>

²www.youtube.com/watch?v=b0XCLR3Wric