UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS CRATEÚS

Resolução de problemas no URI

Marcus Vinícius Martins Melo

INTRODUÇÃO

Problema #1

Vamos iniciar pelo o problema 3115 do URI.

URI Online Judge | 3115

Estradas

Por Giovanna Kobus Conrado, University of São Paulo Brazil

Timelimit: 1

Tlândia é um país muito bonito. Ele consiste de **N** cidade conectadas por **N**-1 estradas bidirecionais de tal for que há um caminho entre qualquer par de cidades. Houve uma enorme tempestade e agora capital encontra-se sem comida para ajudar seus cidadãos.

Cada uma das outras cidades ofereceu enviar um caminhão de comida para ajuda, mas há um problema: cada estrada de Tlândia tem um limite máximo de peso dos caminhões que podem passar por ela.

Agora o governo de Tlândia quer saber qual o peso do caminhão mais pesado que cada uma das cidades pode mandar para a capital sem exceder o limite de peso de nenhuma estrada.

Entrada

A primeira linha da entrada consiste de um único inteiro **N**, o número de cidades em Tlândia ($1 \leq N \leq 10^5$). As cidades são numeradas de 1 a **N**.

As **N**-1 linhas seguintes consistem de três inteiros cada: **V**, **U** e **C**, indicando que existe uma estrada entre a cidade **V** e a cidade **U** que aguenta caminhões que pesam até **C** quilos ($1 \leq V,U \leq N$ e $1 \leq C \leq 10^9$).

A capital será sempre a cidade representada por **N**.

Saída

A saída deve ser uma linha contendo **N**-1 inteiros, o **i**-ésimo dos quais é o peso do caminhão mais pesado que a cidade **i** consegue mandar para a capital sem exceder o limite de peso de nenhuma estrada.

Exemplo de Entrada	Exemplo de Saída
5 3 5 3 2 4 1 3 2 4 1 4 1	1 3 3 1

Problema #2

Vamos ao problema 1531 do URI.

URI Online Judge | 1531

Fibonacci de Novo!

Por Gabriel Dalalio, ITA Brazil

Timelimit: 3

A famosa sequência de Fibonacci pode ser definida da seguinte maneira:

- Fib(1) = Fib(2) = 1
- Fib(N) = Fib(N-1) + Fib(N-2), para N > 2

Sua tarefa é simples, calcular o valor do resto de Fib(Fib(N)) por M.

Entrada

A entrada é composta por vários casos de teste e termina com EOF. Cada caso de teste consiste de uma linha com dois inteiros **N** e **M** ($1 \leq$ **N** $\leq 10^9$, $2 \leq$ **M** $\leq 10^6$).

Saída

Para cada caso de teste, imprima uma linha contendo um inteiro igual ao resto de Fib(Fib(**N**)) por **M**.

Fibonacci

Os números de fibonacci são uma sequencia de números definida pela seguinte equação de recorrência:

$$F_n = \begin{cases} 0, & \text{se } n = 0. \\ 1, & \text{se } n = 1. \\ F_{n-1} + F_{n-2}, & \text{se } n \geq 2. \end{cases}$$

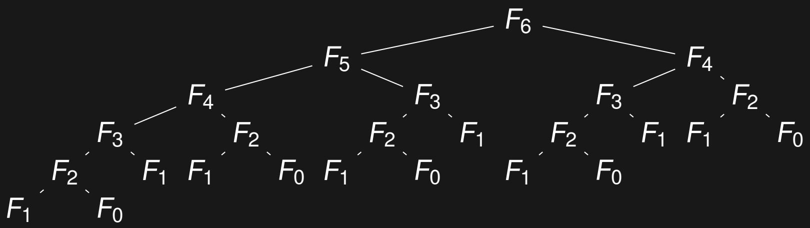
Exemplos:

$$F_2 = F_1 + F_0 = 1.$$
$$F_3 = F_2 + F_1 = 2.$$
$$\ldots$$

```
int fib(int n) {
	if (n == 0) return 0;
	if (n == 1) return 1;
	return fib(n - 1) + fib(n - 2);
}
```

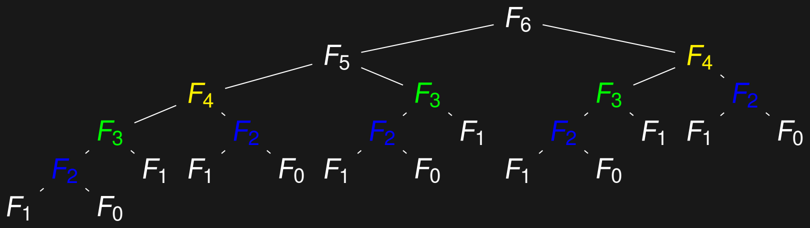
Fibonacci

Arvore de recursão para F_6



Fibonacci

Problema: Vários subproblemas são recalculados desnecessariamente.



Para calcular a complexidade de tempo do algoritmo tomamos a seguinte relação de recorrência:

$$T(n) = T(n-1) + T(n-2) + c.$$

Para simplificar o calculo suponha que:

$$T'(n) = 2T(n-1) + c$$
$$T''(n) = 2T(n-2) + c$$

Perceba que $T'(n) \geq T(n) \geq T''(n)$.

Utilizando o método da soma telescopia temos que:

$$\begin{aligned} T'(n) &= 2T(n-1) + c \\ &= 2 \cdot 2T(n-2) + c \\ &= 2 \cdot 2 \cdot 2T(n-3) + c \\ &\dots \\ &= 2^k T(n-k) + c \end{aligned}$$

Vamos encontrar o valor de k para qual, $n - k = 0, k = n$. Assumindo $T(0) = 1$.

$$T'(n) = 2^n T(0) + c$$

Logo temos: $T'(n) \approx O(2^n)$.

Similarmente para $T''(n)$ temos que:

$$\begin{aligned} T''n) &= 2T(n-2) + c \\ &= 2 \cdot 2T(n-4) + c \\ &= 2 \cdot 2 \cdot 2T(n-6) + c \\ &\cdots \\ &= 2^k T(n-2k) + c \end{aligned}$$

Vamos encontrar o valor de k para qual, $n - 2k = 0, k = \frac{n}{2}$. Assumindo $T(0) = 1$.

$$T''(n) = 2^{\frac{n}{2}} T(0) + c$$

Logo temos: $T''(n) \approx O(2^{\frac{n}{2}})$

Lembrando que como $T'(n) \geq T(n) \geq T''(n)$, então:

$$O(2^n) \geq T(n) \geq O(2^{\frac{n}{2}}).$$

Vamos fazer algumas modificações:

$$F_n = F_{n-1} + F_{n-2}$$

$$F_n = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix}$$

Fibonacci

Exponenciação por matrizes

Exemplos:

$$F_2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} \\ F_1 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} \\ F_1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} \end{bmatrix}$$

Vamos fazer algumas modificações:

$$F_n = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix}$$

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix}$$

$$\begin{bmatrix} F_2 \\ F_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$\begin{bmatrix} F_3 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_2 \\ F_1 \end{bmatrix}$$

$$\begin{bmatrix} F_3 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$\begin{bmatrix} F_3 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$\begin{bmatrix} F_4 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n-1} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

Por conveniência utilizaremos:

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

Agora assumimos que $F_0 = 0$ e $F_1 = 1$:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix}$$

Similarmente, trocando n por 2n, temos:

$$\begin{bmatrix} F_{2n+1} \\ F_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{2n} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$\begin{bmatrix} F_{2n+1} \\ F_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} F_{2n+1} \\ F_{2n} \end{bmatrix} = \begin{bmatrix} (F_{n+1})^2 + (F_n)^2 \\ F_n \cdot F_{n+1} + F_{n-1} \cdot F_n \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Substituindo $F_{n-1} = F_{n+1} - F_n = F_n + F_{n-1} - F_n = F_{n-1}$ e fazendo algumas modificações, temos:

$$\begin{bmatrix} F_{2n+1} \\ F_{2n} \end{bmatrix} = \begin{bmatrix} (F_{n+1})^2 + (F_n)^2 \\ 2F_{n+1} \cdot F_n - (F_n)^2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Perceba que:

$$2^{84} = 2^{64} \cdot 2^{16} \cdot 2^{4}$$

Similarmente temos que:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{13} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{8} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{4} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{1}$$

$$(13)_{10} = (1101)_2$$

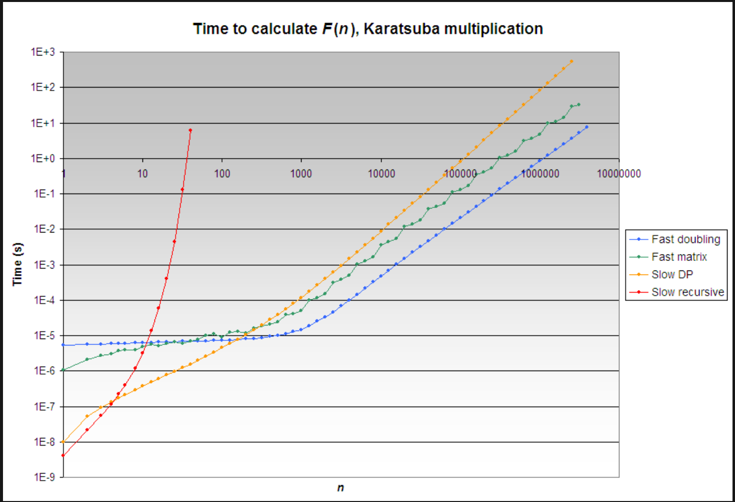
Método $O(logn)$.

Formula fechada para calculo do fibonacci de n, $O(1)$.

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

Problema: Para valores muito grandes pode haver problemas com a precisao.

Fibonacci
Analise dos métodos



Fibonacci

Voltando a questão:

URI Online Judge | 1531

Fibonacci de Novo!

Por Gabriel Dalalio, ITA Brazil

Timelimit: 3

A famosa sequência de Fibonacci pode ser definida da seguinte maneira:

- Fib(1) = Fib(2) = 1
- Fib(N) = Fib(N-1) + Fib(N-2), para N > 2

Sua tarefa é simples, calcular o valor do resto de Fib(Fib(N)) por M.

Entrada

A entrada é composta por vários casos de teste e termina com EOF. Cada caso de teste consiste de uma linha com dois inteiros **N** e **M** ($1 \le N \le 10^9$, $2 \le M \le 10^6$).

Saída

Para cada caso de teste, imprima uma linha contendo um inteiro igual ao resto de Fib(Fib(**N**)) por **M**.

E onde entra o modulo?:

I	0	1	2	3	4	5	6	7	8	9	10	11
F_i	0	1	1	2	3	5	8	13	21	34	55	89
F_i *mod 2*	0	1	1	0	1	1	0	1	1	0	1	1
F_i *mod 3*	0	1	1	2	0	2	2	1	0	1	1	2

Para $m = 2$ o período é 011 e tem tamanho 3, já para $m = 3$ o periodo tem tamanho 8.

Utilizando o Pisano Period, temos que:

$$
\begin{gathered}
fib(2019) \bmod 3 \equiv fib(2019 \ mod \ 8) \bmod 3, \\
fib(fib(n)) \ mod \ m \equiv fib(fib(n \ mod \ P_m)) \ mod \ m
\end{gathered}
$$

Vamos programar!!!

Vamos ao problema 2501 do URI.

Fatores permitidos

URI Online Judge | 2501

Fatores Permitidos

Por Francisco Elio Parente Arcos Filho, UEA Brazil

Timelimit: 3

Professor Chico, suspeitando que Levi, seu aluno, não está estudando Programação dinâmica como deveria, resolveu tramar um plano para incentivar Levi a estudar mais.

Chico avisou aos seus alunos que agora eles seriam referenciados por códigos numéricos de no máximo 12 dígitos nas comunicações oficiais (e-mails e tarefas). E logo em seguida entregou a cada, um cartão que continha um único número escrito. Rapidamente os alunos presumiram que esse seria seu código, mas para surpresa dos alunos e desespero de Levi, professor chico explicou que esses não eram seus códigos.

O código de um aluno era o termo de uma sequência ordenada **S** que estava na posição (indexada a partir de 1) especificada pelo número no cartão de cada um. Essa sequência tem uma característica especial: cada termo, quando decomposto em fatores primos, só pode ter números contidos em um conjunto de **N** elementos escritos no quadro pelo professor. E pra dificultar ainda mais a vida de Levi, esses números mudariam toda semana de tal forma que ele sempre terá de recalcular seu código se não quiser atrasar suas tarefas.

Sua tarefa é fazer um programa para ajudar Levi de tal modo que, dado os números primos escritos no quadro durante a semana pelo professor Chico e número no cartão, diga o seu código semanal.

Fatores permitidos

Entrada

A entrada é composta de vários casos de testes. A primeira linha de um caso de teste contém dois números inteiros **N** ($1 \leq N \leq 10^2$) e **M** ($1 \leq M \leq 10^5$) representando respectivamente a quantidade de números escritos no quadro pelo professor Chico e o número escrito no cartão. A segunda linha contém **N** números primos P_i ($2 \leq P_i < 10^6$) ordenados de forma crescente, onde P_i ($1 \leq i \leq N$) é um número escrito no quadro. A entrada termina quando **N**=**M**=0.

Saída

A saída consiste de uma linha por caso de teste contendo o código semanal de Levi.

Exemplo de Entrada	Exemplo de Saída
2 1 2 3 2 10 2 3 3 10 2 3 5 3 10 3 7 13 0 0	2 24 15 81

Problema #4

Vamos ao problema 2076 do URI.

Alocação commodities

URI Online Judge | 2076

Alocação Ótima de Commodities

Por XII Maratona de Programação IME-USP, 2008 Brazil

Timelimit: 1

Tjalling C. Koopmans ganhou em 1975 o prêmio Nobel de Economia juntamente com o matemático russo Kantorovich pelas suas contribuições em importantes áreas como a alocação ótima de recursos. Koopmans formou-se em Matemática pela Universidade de Utrecht, na Holanda, e se especializou em economia matemática. Durante a segunda guerra mundial esteve envolvido no estudo de alocação ótima de recursos, que 30 anos mais tarde lhe rendeu o prêmio Nobel. É considerado um dos precursores da teoria de programação linear. Suas contribuições têm importantes aplicações em Economia, Matemática, Física e mesmo em Química.

Um dos problemas prediletos de Koopmans era o de alocação ótima de commodities. Neste problema, é dado um valor inicial e um valor final da aplicação a ser feita. Entretanto, nem todos os valores podem ser aplicados nos vários investimentos. Cada investimento é definido através de um número inteiro, e, por convenção, apenas quando o valor a ser aplicado for um múltiplo de pelo menos um número que define um investimento ele pode ser aplicado.

Sua tarefa neste problema é calcular o valor máximo que pode ser aplicado. Ou seja, dado o valor inicial e valor final a serem aplicados e uma lista de inteiros que definem as várias aplicações,você deverá calcular a soma dos valores que podem ser aplicados no intervalo.

Alocação commodities

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro **T** indicando o número de instâncias.

A primeira linha de cada instância possui três inteiros **I**, **F** e **N** ($1 < I < F < 1000000000$ e $1 < N < 20$) que representam o valor inicial, o valor final e o número de elementos da lista de aplicações. A próxima linha contém **N** inteiros $1 < a_i < 1000000000$ indicando a lista de aplicações.

Saída

Para cada instância imprima uma linha contendo a soma dos valores que podem ser aplicados no intervalo. Como este valor pode ser muito grande então imprima o resultado módulo 1300031.

Exemplo de Entrada	Exemplo de Saída
3 1 10 1 1 1 9 2 3 5 1 999 2 3 5	55 23 233168

REFERÊNCIAS

Referências

CHAKRABORTY. kushal. *Fast Doubling method*. 2020. Disponível em: <https://www.geeksforgeeks.org/fast-doubling-method-to-find-the-nth-fibonacci-number/>.

FAST Doubling method. Vinay Kumar, 2016. Disponível em: <https://www.hackerearth.com/pt-br/practice/notes/fast-doubling-method-to-find-nth-fibonacci-number/>.

PROJECT Nayuki. nayuki, 2015. Disponível em: <https://www.nayuki.io/page/fast-fibonacci-algorithms>.

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS CRATEÚS

Resolução de problemas no URI

Marcus Vinícius Martins Melo