Recorrências e Combinatória

Notas de aula da disciplina
TE: Técnicas de Construção de Algoritmos

Fabiano de Souza Oliveira
(fabiano.oliveira@ime.uerj.br)

Paulo Eustáquio Duarte Pinto
(pauloedp arroba ime.uerj.br)

agosto/2020

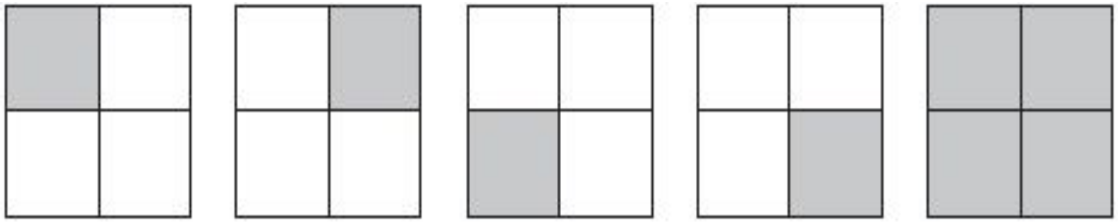
TE: Técnicas de Construção de Algoritmos
Recorrências e Combinatória

Problemas de 08/08/2020:

1323 - Feynman
2793 - HM
1946 - Pirâmide da Sorte
2777 - Subsets do Dabriel
1474 - Ônibus

1323 - Feynman

1. **Contexto:** Feynman foi um físico famoso que gostava de quebra-cabeças. Recentemente encontraram um antigo desafio proposto por ele: dado um quadriculado **n x n**, quantos quadrados distintos existem no total?



2. **Entrada:** Uma série de casos de testes descritos cada um em uma linha, terminados com **0**, que não deve ser processado. Cada linha contém um inteiro **n** (1 ≤ **n** ≤ 100).

3. **Saída:** Para cada teste deve ser impresso o número de quadrados distintos.

4. **Exemplo de entrada:**

```
2
1
8
0
```

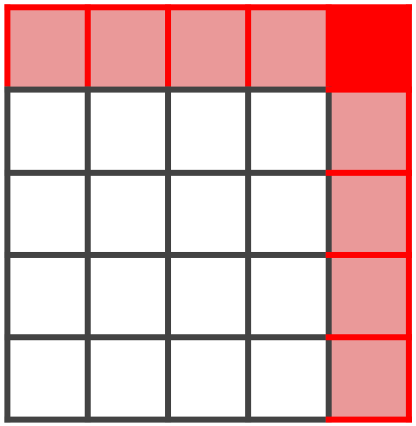
Exemplo de saída:

```
5
1
204
```

1323 - Feynman

Dicas:

1. Escrever e solucionar a recorrência



2. Contar os novos quadrados quando a dimensão aumenta de $n-1$ para n:

- quadrados de lado 1: $2(n-1)+1 = 2n-1$
- . . .
- quadrado de lado n: 1

2793 - HM

1. **Contexto:** Em uma loja, roupas de homem (H) e de mulher (M) estão enfileiradas, havendo o mesmo número de H e M. Isabel quer particionar a fila, de forma que cada partição tenha o mesmo número de H e M. De quantas maneiras ela pode fazer isso?

2. **Entrada:** Uma única entrada contendo um string com tamanho entre 2 e 10^7, contendo apenas H e M, em igual quantidade.

3. **Saída:** Deve ser impresso um inteiro, em módulo 1.000.000.007, indicando a quantidade de partições distintas.

4. **Exemplo de entrada 1:**
MHMH

Exemplo de saída 1:
2

Exemplo de entrada 2:
MHMHHM

Exemplo de saída 2:
4

Exemplo de entrada 3:
MMHH

Exemplo de saída 3:
1

2793 - HM

Dicas:

1. **Trabalhar com inteiros de 64 bits**

Exemplo: MHHMHMHHMHMM

1 partição: MHHMHMHHMHMM

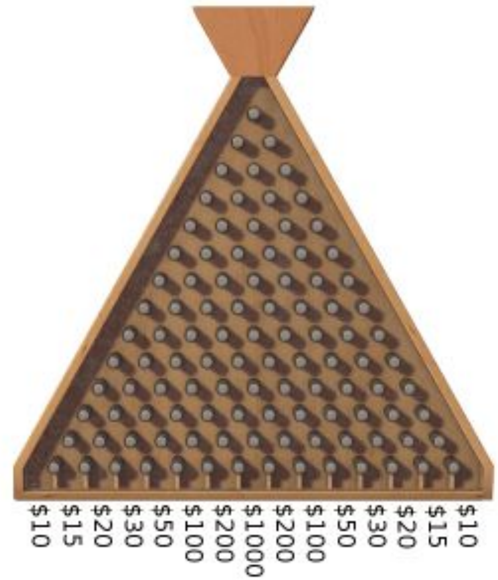
2 partições: MH HMHMHHMHMM
MHHM HMHHMHMM
MHHMHM HHMHMM

3 partições: MH HM HMHHMHMM
.....

4 partições: MH HM HM HHMHMM

1946 - Pirâmide da Sorte

1. **Contexto:** Um show de TV sorteia prêmios através de um mecanismo onde uma bolinha desce uma pirâmide até a base, com número ímpar de posições. A cada posição intermediária, ela pode desviar p/ esquerda ou direita com probabilidade 0.5. Quer-se saber a probabilidade dela chegar no ponto central.



2. **Entrada:** Um único caso indicado p/ um inteiro **S** (3 ≤ **S** ≤ 4999), um número ímpar indicando o número de posições na base.

3. **Saída:** Um único real com duas posições decimais indicando a probabilidade pedida.

4. **Exemplo de entrada 1:**
3

Exemplo de saída 1:
50.00

Exemplo de entrada 2:
5

Exemplo de saída 2:
37.50

Exemplo de entrada 3:

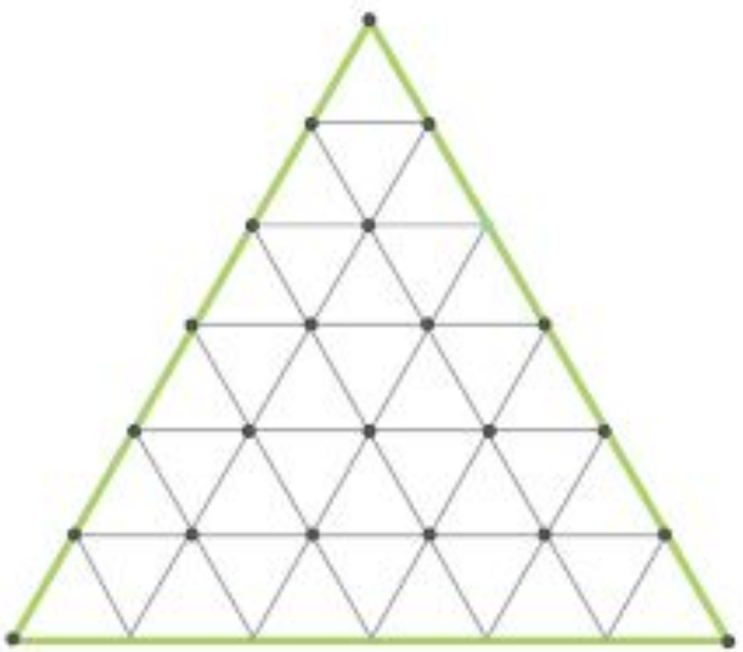
Exemplo de saída 3:

1946 - Pirâmide da Sorte

Dicas:

1. Trabalhar com double (reais)

2. Representar posições da pirâmide numa matriz



	0	0	0	0	0	0
		0	0	0	0	0
			0	0	0	0
				0	0	0
					0	0
						0

3. Usar memorização

1946 - Pirâmide da Sorte

Dica 4: Recorrência, baseada em probabilidade condicional:

T[1,1] = 1; **T[i,j]=0 à direita da diagonal principal**

T[i,j] = (soma das probabilidades das posições anteriores que alcançam (i,j))*0.5

1	0	0	0	0	0	0
		0	0	0	0	0
			0	0	0	0
				0	0	0
					0	0
						0

1946 - Pirâmide da Sorte

Dica 5: Imprimir com quantidade fixa de decimais em C++

```
#include<iomanip>
```

```
cout<<fixed<<setprecision(2)<<x;
```

2777 - Subsets do Dabriel

1. **Contexto:** Dabriel inventou um quebra-cabeças que consiste em, dados os inteiros 1 a N, encontrar o número de subconjuntos MAXIMAIS dos números 1 a N tal que, se i está no subconjunto, nem i-1 nem i+1 podem também estar. Para i = 5 existem 4 subconjuntos: {1, 3, 5}, {1, 4}, {2, 4}, {2, 5}.

EXEMPLOS:

N = 1: {1}

N = 2: {1} {2}

N = 3: {1, 3} {2}

N = 4: {1, 3} {1, 4} {2, 4}

N = 5: {1, 3, 5} {1, 4} {2, 4} {2, 5}

N = 6: ?

2777 - Subsets do Dabriel

1. **Contexto:** Dabriel inventou um quebra-cabeças que consiste em, dados os inteiros 1 a N, encontrar o número de subconjuntos MAXIMAIS dos números 1 a N tal que, se i está no subconjunto, nem i-1 nem i+1 podem também estar. Para i = 5 existem 4 subconjuntos: {1, 3, 5}, {1, 4}, {2, 4}, {2, 5}.

2. **Entrada:** Uma série de casos de testes, terminados por fim de arquivo. Cada teste em numa linha com um inteiro N, ($1 \leq N \leq 10^{18}$).

3. **Saída:** Para cada teste deve ser impresso o número de subconjuntos maximais que podem ser formados, em módulo 1.000.000.007.

4. **Exemplo de entrada:**

```
1
5
30
```

Exemplo de saída:

```
1
4
4410
```

2777 - Subsets do Dabriel

Dicas:

1. N = 1: {1}
   N = 2: {1} {2}
   N = 3: {1, 3} {2}
   N = 4: {1, 3} {1, 4} {2, 4}
   N = 5: {1, 3, 5} {1, 4} {2, 4} {2, 5}
   N = 6: ?

2. Trabalhar com inteiros de 64 bits

3. Idéia da recorrência de contagem: para n > 3, pensar em quantos subconjuntos têm o número n-1 e quantos têm n (são mutuamente exclusivos e englobam todos os casos)

4. Recorrência:

T[1] = 1; T[2] = 2; T[3] = 2;
T[n] = T[n-2] + T[n-3], se n > 3

2777 - Subsets do Dabriel

4. Implementação recorrência linear: $T_n = 0.T_{n-1} + 1.T_{n-2} + 1.T_{n-3}$

$$\underbrace{\begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}}_{A} \times \underbrace{\begin{vmatrix} T_3 \\ T_2 \\ T_1 \end{vmatrix}}_{B} = \underbrace{\begin{vmatrix} T_4 \\ T_3 \\ T_2 \end{vmatrix}}_{C} \quad A^{n-3} \times B = D = \begin{vmatrix} T_n \\ T_{n-1} \\ T_{n-2} \end{vmatrix}$$

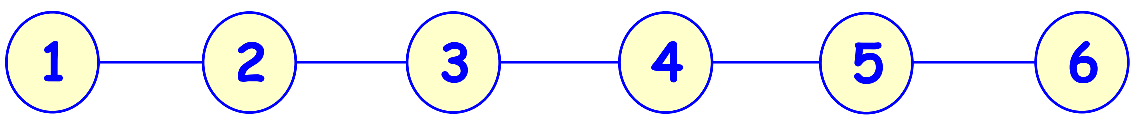
$T_n = A^{n-3}[1,1].T_3 + A^{n-3}[1,2].T_2 + A^{n-3}[1,3].T_1 =$

$A^{n-3}[1,1].2 + A^{n-3}[1,2].2 + A^{n-3}[1,3].1$

2777 - Subsets do Dabriel

Curiosidade: este problema pode ser formulado em termos de grafos:

Encontrar o número de conjuntos independentes maximais de um caminho P_n



Conjuntos independentes maximais de P_6:

{1,3,5} {1, 3, 6} {1, 4, 6} {2, 4, 6} {2, 5}

1474 - Ônibus

Contexto: Tem-se um estacionamento em linha reta de comprimento N. Quer-se saber de quantas maneiras distintas pode-se preencher totalmente essa linha com ônibus de comprimento 10 ou micro-ônibus, de comprimento 5. Os micro-ônibus podem ter K cores distintas e os ônibus, L cores distintas. Se duas possibilidades variam apenas na cor de algum dos veículos, a configuração é considerada distinta.

Entrada: Uma série de casos de testes, terminados por fim de arquivo. Cada caso de teste vem em uma linha, com 3 inteiros: N ($5 \leq N \leq 10^{15}$, múltiplo de 5), K ($5 \leq K \leq 10^{15}$), e L ($5 \leq L \leq 10^{15}$).

Saída: Para cada teste deve ser impresso o número de possibilidades de formar a linha, expresso em módulo 1.000.000.

Exemplo de entrada:

```
25 5 5
5 1000 1000
20 17 31
15 9 2
```

Exemplo de saída:

```
006000
001000
111359
000765
```

1474 - Ônibus

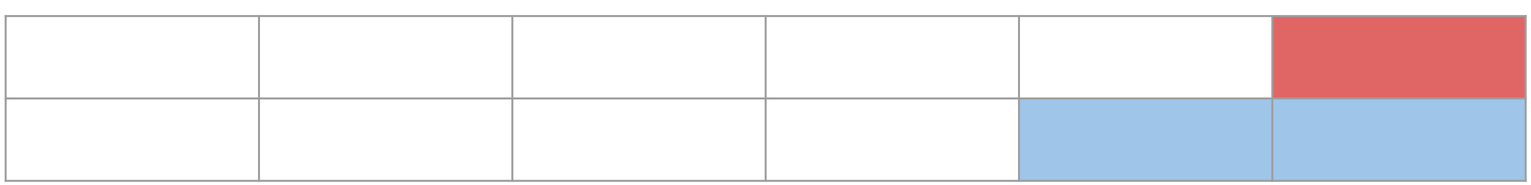
Dicas:

1. Usar inteiros de 64 bits

2. Observar a saída com zeros à esquerda.

3. Dividir os comprimentos por 5: N/5, 2, 1, para o tamanho do estacionamento, dos ônibus e dos microônibus, respect.

4. Formulação da recorrência: pensar que, para N > 1, a fila do estacionamento pode terminar de duas maneiras: com um micro-ônibus ou um ônibus.



5. Resolver com exponenciação rápida de matrizes.

FIM