

Algoritmos e Programação III

Trabalho I

Matthias Oliveira de Nunes

10 de outubro de 2013

Resumo

Este artigo descreve alternativas de solução para o primeiro trabalho proposto pela disciplina de Algoritmos e Programação III, que consiste em descobrir o número mínimo de amoras possível para nove tripulantes. Esse número, para ser válido, precisa sofrer algumas operações e no final sua divisão pelos tripulantes deve ter resto zero. Serão apresentadas alternativas de solução, como foram implementadas e seus desempenhos na hora de execução.

1 Introdução

Dentro do escopo da disciplina de Algoritmos e Programação III, o primeiro problema pode ser resumido como o seguinte: nove tripulantes pousaram em um planeta coberto por amoreiras, passaram o dia colhendo amoras, jantaram e foram dormir. No meio da noite, um dos tripulantes acorda, divide as amoras igualmente, rouba sua parte, se livra do resto da divisão e volta a dormir. Cada um dos outros tripulantes acorda e faz a mesma coisa, dividindo as amoras restantes igualmente e escondendo sua parte. Curiosamente, sempre sobra o mesmo número de amoras em todas as divisões. Pela manhã, os tripulantes dividem o que restou, mas dessa vez não teve resto na divisão.

Ao final desse enunciado, foram levantadas 3 perguntas:

1. Qual o menor número possível de amoras na pilha sabendo que são nove tripulantes e a sobra de amoras é sempre igual?
2. Qual a quantidade de amoras na manhã seguinte?

3. Quantas amoras cada um dos tripulantes receberá?

Junto a isso, foi pedido que fosse fornecido resultado para seis casos diferentes, com as sobras variando de 1 a 6. Se possível, também obter resultados para 6, 7, 8, 10, 11 ou mais tripulantes.

Para que fique mais claro o processo que o número deve sofrer para ser válido, será demonstrado que 25, tendo 1 como resto das divisões, para uma tripulação de 3 integrantes, é um número de amoras possível.

$$\begin{array}{r} 25 \mid 3 \\ 1 \mid 8 \end{array} \quad \begin{array}{r} 25 \\ - 1 \\ \hline 24 \end{array} \quad \begin{array}{r} 24 \\ - 8 \\ \hline 16 \end{array}$$
$$\begin{array}{r} 16 \mid 3 \\ 1 \mid 5 \end{array} \quad \begin{array}{r} 16 \\ - 1 \\ \hline 15 \end{array} \quad \begin{array}{r} 15 \\ - 5 \\ \hline 10 \end{array}$$
$$\begin{array}{r} 10 \mid 3 \\ 1 \mid 3 \end{array} \quad \begin{array}{r} 10 \\ - 1 \\ \hline 9 \end{array} \quad \begin{array}{r} 9 \\ - 3 \\ \hline 6 \end{array}$$
$$\begin{array}{r} 6 \mid 3 \\ 0 \mid 2 \end{array}$$

Como o resto sempre se manteve 1 e na quarta divisão foi para 0, logo esse número é válido.

Para resolver o problema proposto, analisaremos duas possíveis alternativas de solução, bem como suas características, optando pela segunda já que a primeira não se aplica a todos os casos. Em seguida os resultados obtidos serão apresentados, bem como as conclusões obtidas no decorrer do trabalho.

2 Primeira Solução

Depois de considerar o problema, podemos perceber que a solução mais óbvia seria utilizar um algoritmo "força bruta", isso é, ir testando de um em um até

encontrar o número desejado. Para cada novo número de amoras, ele tenta fazer as divisões, sempre retirando a nona parte junto com o resto. Se o resto não se mantiver igual ele pula para tentar outro número, e se na última divisão o resto não for zero, ele também pula para tentar outro número.

Tendo isso como base, um algoritmo implementando esta idéia seria parecido com este:

```
pronto = FALSO
tripulantes = 9;

para amoras de tripulantes ate pronto = VERDADEIRO
{
    amorasPelaManha = amoras;
    restoConstante = amoras % tripulantes;

    se (restoConstante != 0)
        para i = 1 ate tripulantes
        {
            restoASerDesperdicado = amorasPelaManha % tripulantes;

            se (restoASerDesperdicado != restoConstante)
                break;

            porcaoIndividual = amorasPelaManha / tripulantes;
            amorasPelaManha = amorasPelaManha - restoASerDesperdicado -
                porcaoIndividual;

            se ((i == tripulantes) && (amorasPelaManha % tripulantes == 0))
                pronto = VERDADEIRO;
        }
}
```

Implementando esse algoritmo foi observado os seguintes problemas:

1. Extremamente lento, já que vai de um em um até encontrar o número.
2. Ele vai sempre achar o número mínimo de amoras, não podendo assim variar o resto de 1 a 6 como havia sido solicitado.

Com base nessas observações, essa solução não é o suficiente para coletar

todos os dados desejados, mas com ela já foi possível constar esses valores:

Tripulantes	Resto	Mínimo de Amoras	Amoras pela manhã	Tempo
6	5	46.631	15.600	0ms
7	1	823.537	279.930	16ms
8	7	16.777.167	5.764.752	277ms
9	1	387.420.481	134.217.720	6379ms

Tabela 1: Força Bruta

**O algoritmo foi implementado em C++ e rodou em cima de um Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50GHz

Já podemos identificar um padrão apenas olhando essa tabela:

1. Números ímpares de tripulantes possuem a menor quantidade de amoras com resto um.
2. Números pares de tripulantes possuem a menor quantidade de amoras com o resto sendo igual a quantidade de tripulantes menos um.

Como esse algoritmo não é capaz de nos dar todas as informações necessárias, foi necessário pensar de uma maneira diferente para chegar a uma solução eficiente.

3 Segunda Solução

Uma vez que a alternativa mais simples não consegue ser flexível o suficiente para produzir todos os dados, optamos por desenvolver uma abordagem mais cuidadosa, experimentando fazer o caminho inverso ao problema. Com aqueles resultados também se tornou possível conferir se o novo algoritmo deu certo ou não.

3.1 Definição

O número de amoras pela manhã, obrigatoriamente deve ser um múltiplo do número de tripulantes, já que o resto da divisão final deve ter resto zero.

Com isso podemos definir esse número como

$$A = i.T$$

onde A é o número de amoras pela manhã, i seria um contador que começaria em 1 e serve para multiplicar T que é a quantidade de tripulantes.

Com esse número inicial, já é o suficiente para achar o próximo valor, que seria simplesmente adicionar a parcela roubada pelo último tripulante junto com o resto descartado. Essa parcela roubada pode ser definida por

$$P = \frac{A}{(T - 1)}$$

onde P é a parcela, e ela *precisa* ser um número inteiro. Vamos definir o resto como R , e somado com a parcela ao número de amoras, se encontra o número antes do ultimo furto.

$$A = A + \frac{A}{(T - 1)} + R$$

Se esse expressão for repetida pelo número de tripulantes, considerando que a multiplicação inicial pelo contador já obteve o valor correto, resultaria no número mínimo de amoras possível para um determinado resto R .

É exatamente isso que o nosso algoritmo irá fazer, para cada incremento no contador, ele vai iterar a definição do próximo número enquanto as parcelas resultarem em números inteiros, e repetirá pelo número de tripulantes.

3.2 Algoritmo

```
tripulacao = 9;
resto = 1;

para i ate infinito
{
    amoras = i * tripulacao;
    amorasManha = amoras;
    pronto = VERDADEIRO;

    para t ate tripulacao
```

```

{
    se ((amoras % (tripulacao - 1)) != 0)
    {
        pronto = FALSO;
        break;
    }

    parcela = amoras / (tripulacao - 1);
    amoras = amoras + parcela + resto;
}

se (pronto = VERDADEIRO)
{
    break;
}
}

```

Pode ser observado a existência de uma variável **amorasManha**, que guarda o ultimo valor de **amoras** antes da iteração. Com isso, no final da execução, se obtém ambos os valores, amoras pela manhã e o número mínimo de amoras possível.

Foi pedido na especificação a quantidade que cada tripulante recebe. Com o número já encontrado, é muito simples de obter essas informações. Um laço apenas fazendo o processo inverso do algoritmo e mandando sair na tela, garante os dados.

```

para j ate tripulacao
{
    print("T" + j + ": " + (amoras/tripulacao +
        amorasManha/tripulacao));
    amoras = amoras - amoras/tripulacao - resto;
}

```

Um exemplo de saída, com 9 tripulantes e resto se mantendo em 1:

```

T1: 57959800
T2: 53176831
T3: 48925303

```

T4: 45146167
T5: 41786935
T6: 38800951
T7: 36146743
T8: 33787447
T9: 31690295

4 Resultados

O algoritmo foi implementado em Java e rodou em cima de um Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50GHz, foram obtidos os seguintes resultados:

Tripulantes	Resto	Mínimo de Amoras	Amoras pela manhã	Tempo
6	1	233.275	78.120	0ms
	2	186.614	62.490	1ms
	3	139.953	46.860	1ms
	4	93.292	31.230	0ms
	5	46.631	15.600	2ms
7	1	823.537	279.930	4ms
	2	1.647.074	559.860	6ms
	3	2.470.611	839.790	3ms
	4	3.294.148	1.119.720	0ms
	5	4.117.685	1.399.650	1ms
	6	4.941.222	1.679.580	1ms
8	1	117.440.505	40.353.600	22ms
	2	100.663.282	34.588.792	28ms
	3	83.886.059	28.823.984	14ms
	4	67.108.836	23.059.176	11ms
	5	50.331.613	17.294.368	9ms
	6	33.554.390	11.529.560	6ms
9	1	387.420.481	134.217.720	34ms
	2	774.840.962	268.435.440	77ms
	3	1.162.261.443	402.653.160	100ms
	4	1.549.681.924	536.870.880	135ms
	5	1.937.102.405	671.088.600	163ms
	6	2.324.522.886	805.306.320	198ms
10	1	89.999.999.991	31.381.059.600	8222ms
	2	79.999.999.982	27.894.275.190	8203ms
	3	69.999.999.973	24.407.490.780	7158ms
	4	59.999.999.964	20.920.706.370	6143ms
	5	49.999.999.955	17.433.921.960	5121ms
	6	39.999.999.946	13.947.137.550	4108ms
11	1	285.311.670.601	99.999.999.990	22316ms
	2	570.623.341.202	199.999.999.980	55752ms
	3	855.935.011.803	299.999.999.970	83783ms
	4	1.141.246.682.404	399.999.999.960	113216ms
	5	1.426.558.353.005	499.999.999.950	140287ms
	6	1.711.870.023.606	599.999.999.940	169716ms

Tabela 2: Mínimo de amoras

Resto	Tripulante	Quantidade de amoras por tripulante
1	T1	51.899
	T2	45.419
	T3	40.019
	T4	35.519
	T5	31.769
	T6	28.644
2	T1	41.517
	T2	36.333
	T3	32.013
	T4	28.413
	T5	25.413
	T6	22.913
3	T1	31.135
	T2	27.247
	T3	24.007
	T4	21.307
	T5	19.057
	T6	17.182
4	T1	20.753
	T2	18.161
	T3	16.001
	T4	14.201
	T5	12.701
	T6	11.451
5	T1	10.371
	T2	9.075
	T3	7.995
	T4	7.095
	T5	6.345
	T6	5.720

Tabela 3: Amoras para 6 tripulantes

Resto	Tripulante	Quantidade de amoras por tripulante
1	T1	157.638
	T2	140.831
	T3	126.425
	T4	114.077
	T5	103.493
	T6	94.421
	T7	86.645
2	T1	315.276
	T2	281.662
	T3	252.850
	T4	228.154
	T5	206.986
	T6	188.842
	T7	173.290
3	T1	472.914
	T2	422.493
	T3	379.275
	T4	342.231
	T5	310.479
	T6	283.263
	T7	259.935
4	T1	630.552
	T2	563.324
	T3	505.700
	T4	456.308
	T5	413.972
	T6	377.684
	T7	346.580
5	T1	788.190
	T2	704.155
	T3	632.125
	T4	570.385
	T5	517.465
	T6	472.105
	T7	433.225
6	T1	945.828
	T2	844.986
	T3	758.550
	T4	684.462
	T5	620.958
	T6	566.526
	T7	519.870

Tabela 4: Amoras para 7 tripulantes

Tripulante	Quantidade de amoras por tripulante
T1	19.724.263
T2	17.889.255
T3	16.283.623
T4	14.878.695
T5	13.649.383
T6	12.573.735
T7	11.632.543
T8	10.809.000

Tabela 5: Amoras para 8 tripulantes com resto 1

Tripulante	Quantidade de amoras por tripulante
T1	16.906.509
T2	15.333.645
T3	13.957.389
T4	12.753.165
T5	11.699.469
T6	10.777.485
T7	9.970.749
T8	9.264.855

Tabela 6: Amoras para 8 tripulantes com resto 2

Tripulante	Quantidade de amoras por tripulante
T1	140.887.55
T2	127.780.35
T3	116.311.55
T4	10.627.635
T5	9.749.555
T6	8.981.235
T7	8.308.955
T8	7.720.710

Tabela 7: Amoras para 8 tripulantes com resto 3

Tripulante	Quantidade de amoras por tripulante
T1	11.271.001
T2	10.222.425
T3	9.304.921
T4	8.502.105
T5	7.799.641
T6	7.184.985
T7	6.647.161
T8	6.176.565

Tabela 8: Amoras para 8 tripulantes com resto 4

Tripulante	Quantidade de amoras por tripulante
T1	8.453.247
T2	7.666.815
T3	6.978.687
T4	6.376.575
T5	5.849.727
T6	5.388.735
T7	4.985.367
T8	4.632.420

Tabela 9: Amoras para 8 tripulantes com resto 5

Tripulante	Quantidade de amoras por tripulante
T1	5.635.493
T2	5.111.205
T3	4.652.453
T4	4.251.045
T5	3.899.813
T6	3.592.485
T7	3.323.573
T8	3.088.275

Tabela 10: Amoras para 8 tripulantes com resto 6

Tripulante	Quantidade de amoras por tripulante
T1	57.959.800
T2	53.176.831
T3	48.925.303
T4	45.146.167
T5	41.786.935
T6	38.800.951
T7	36.146.743
T8	33.787.447
T9	31.690.295

Tabela 11: Amoras para 9 tripulantes com resto 1

Tripulante	Quantidade de amoras por tripulante
T1	115.919.600
T2	106.353.662
T3	97.850.606
T4	90.292.334
T5	83.573.870
T6	77.601.902
T7	72.293.486
T8	67.574.894
T9	63.380.590

Tabela 12: Amoras para 9 tripulantes com resto 2

Tripulante	Quantidade de amoras por tripulante
T1	173.879.400
T2	159.530.493
T3	146.775.909
T4	135.438.501
T5	125.360.805
T6	116.402.853
T7	108.440.229
T8	101.362.341
T9	95.070.885

Tabela 13: Amoras para 9 tripulantes com resto 3

Tripulante	Quantidade de amoras por tripulante
T1	231.839.200
T2	212.707.324
T3	195.701.212
T4	180.584.668
T5	167.147.740
T6	155.203.804
T7	144.586.972
T8	135.149.788
T9	126.761.180

Tabela 14: Amoras para 9 tripulantes com resto 4

Tripulante	Quantidade de amoras por tripulante
T1	289.799.000
T2	265.884.155
T3	244.626.515
T4	225.730.835
T5	208.934.675
T6	194.004.755
T7	180.733.715
T8	168.937.235
T9	158.451.475

Tabela 15: Amoras para 9 tripulantes com resto 5

Tripulante	Quantidade de amoras por tripulante
T1	347.758.800
T2	319.060.986
T3	293.551.818
T4	270.877.002
T5	250.721.610
T6	232.805.706
T7	216.880.458
T8	202.724.682
T9	190.141.770

Tabela 16: Amoras para 9 tripulantes com resto 6

Tripulante	Quantidade de amoras por tripulante
T1	12.138.105.959
T2	11.238.105.959
T3	10.428.105.959
T4	9.699.105.959
T5	9.043.005.959
T6	8.452.515.959
T7	7.921.074.959
T8	7.442.778.059
T9	7.012.310.849
T10	6.624.890.360

Tabela 17: Amoras para 10 tripulantes com resto 1

Tripulante	Quantidade de amoras por tripulante
T1	10.789.427.517
T2	9.989.427.517
T3	9.269.427.517
T4	8.621.427.517
T5	8.038.227.517
T6	7.513.347.517
T7	7.040.955.517
T8	6.615.802.717
T9	6.233.165.197
T10	5.888.791.429

Tabela 18: Amoras para 10 tripulantes com resto 2

Tripulante	Quantidade de amoras por tripulante
T1	9.440.749.075
T2	8.740.749.075
T3	8.110.749.075
T4	7.543.749.075
T5	7.033.449.075
T6	6.574.179.075
T7	6.160.836.075
T8	5.788.827.375
T9	5.454.019.545
T10	5.152.692.498

Tabela 19: Amoras para 10 tripulantes com resto 3

Tripulante	Quantidade de amoras por tripulante
T1	8.092.070.633
T2	7.492.070.633
T3	6.952.070.633
T4	6.466.070.633
T5	6.028.670.633
T6	5.635.010.633
T7	5.280.716.633
T8	4.961.852.033
T9	4.674.873.893
T10	4.416.593.567

Tabela 20: Amoras para 10 tripulantes com resto 4

Tripulante	Quantidade de amoras por tripulante
T1	6.743.392.191
T2	6.243.392.191
T3	5.793.392.191
T4	5.388.392.191
T5	5.023.892.191
T6	4.695.842.191
T7	4.400.597.191
T8	4.134.876.691
T9	3.895.728.241
T10	3.680.494.636

Tabela 21: Amoras para 10 tripulantes com resto 5

Tripulante	Quantidade de amoras por tripulante
T1	5.394.713.749
T2	4.994.713.749
T3	4.634.713.749
T4	4.310.713.749
T5	4.019.113.749
T6	3.756.673.749
T7	3.520.477.749
T8	3.307.901.349
T9	3.116.582.589
T10	2.944.395.705

Tabela 22: Amoras para 10 tripulantes com resto 6

Tripulante	Quantidade de amoras por tripulante
T1	35.028.333.690
T2	32.670.385.999
T3	30.526.797.189
T4	28.578.080.089
T5	26.806.519.089
T6	25.196.009.089
T7	23.731.909.089
T8	22.400.909.089
T9	21.190.909.089
T10	20.090.909.089
T11	19.090.909.089

Tabela 23: Amoras para 11 tripulantes com resto 1

Tripulante	Quantidade de amoras por tripulante
T1	70.056.667.380
T2	65.340.771.998
T3	61.053.594.378
T4	57.156.160.178
T5	53.613.038.178
T6	50.392.018.178
T7	47.463.818.178
T8	44.801.818.178
T9	42.381.818.178
T10	40.181.818.178
T11	38.181.818.178

Tabela 24: Amoras para 11 tripulantes com resto 2

Tripulante	Quantidade de amoras por tripulante
T1	105.085.001.070
T2	98.011.157.997
T3	91.580.391.567
T4	85.734.240.267
T5	80.419.557.267
T6	75.588.027.267
T7	71.195.727.267
T8	67.202.727.267
T9	63.572.727.267
T10	60.272.727.267
T11	57.272.727.267

Tabela 25: Amoras para 11 tripulantes com resto 3

Tripulante	Quantidade de amoras por tripulante
T1	140.113.334.760
T2	130.681.543.996
T3	122.107.188.756
T4	114.312.320.356
T5	107.226.076.356
T6	100.784.036.356
T7	94.927.636.356
T8	89.603.636.356
T9	84.763.636.356
T10	80.363.636.356
T11	76.363.636.356

Tabela 26: Amoras para 11 tripulantes com resto 4

Tripulante	Quantidade de amoras por tripulante
T1	175.141.668.450
T2	163.351.929.995
T3	152.633.985.945
T4	142.890.400.445
T5	134.032.595.445
T6	125.980.045.445
T7	118.659.545.445
T8	112.004.545.445
T9	105.954.545.445
T10	100.454.545.445
T11	95.454.545.445

Tabela 27: Amoras para 11 tripulantes com resto 5

Tripulante	Quantidade de amoras por tripulante
T1	210.170.002.140
T2	196.022.315.994
T3	183.160.783.134
T4	171.468.480.534
T5	160.839.114.534
T6	151.176.054.534
T7	142.391.454.534
T8	134.405.454.534
T9	127.145.454.534
T10	120.545.454.534
T11	114.545.454.534

Tabela 28: Amoras para 11 tripulantes com resto 6

5 Conclusões

A solução inicial, foi o primeiro passo para que pudéssemos entender o problema em mãos, e serviu para abrir caminho até a solução definitiva. Essa se mostrou muito mais eficiente, embora tenha exigido uma abordagem menos intuitiva já que resolve o problema de trás para frente. O esperto dele, é que

a cada incrementada no contador, dá um salto enorme no número mínimo de amoras possível, que no primeiro algoritmo, ia de um a um.

Embora não tenha sido mostrado nesse artigo, foi feito testes para 11 e 12 tripulantes, o algoritmo não respondeu bem, levando algumas horas para achar a solução.

Acreditamos ter desenvolvido uma solução interessante e simples de ser implementada, porém, não eficiente para grandes quantidades de tripulantes. Gostaríamos de ter tentado achar uma fórmula que representasse o problema. Caso tivesse sido encontrada, o resultado seria imediato para qualquer número de tripulantes.