

E04 Hanoi4

Vencimento: domingo, 10 nov. 2024, 23:59

Estude inicialmente o exercício teórico [S02](#). Neste exercício, você deve escrever um programa chamado Hanoi4.java que resolve $\text{Hanoi4}(N)$ para qualquer $N \geq 1$ dado.

Espera-se que seu programa dê soluções "eficientes", isto é, com poucos movimentos (dentro do possível). Em particular, quando o número de discos N é da forma $\binom{n+1}{2}$ ($n \geq 1$), seu programa deve dar uma solução com número de movimentos não maior que $(n - 1)2^n + 1$.

Seu programa deve ter dois modos de execução, como ilustrado nos dois exemplos a seguir (nestes exemplos, estamos resolvendo o caso $N = 5$).

```
$ java-introcs Hanoi4 5
1 1 2 3 1 3 3 1 4 2 3 2 5 1 3 0 4 1 3 1 1 0 2 1 1 1
$ java-introcs Hanoi4 5 .
13 moves
1 1 2 3 1 3 3 1 4 2 3 2 5 1 3 0 4 1 3 1 1 0 2 1 1 1
$
```

Na primeira execução acima, apenas os movimentos que constituem a solução foram impressos. Na segunda execução (em que um segundo argumento de linha de comando é dado), o número de movimentos na solução foi também impresso. Note que o número de movimentos deve ser impresso antes dos movimentos em si.

Simulador. Para verificar a correção da solução encontrada pelo seu programa, você pode usar o programa Hanoi4Sim.java abaixo. No momento, você não precisa entender completamente como o programa Hanoi4Sim.java funciona (ele usa pilhas, que ainda não estudamos), mas vale a pena você examinar Hanoi4Sim.java. Seguem alguns exemplos de uso de Hanoi4Sim.java.

```
$ java-introcs Hanoi4 2 | java-introcs Hanoi4Sim 2 .
Initial configuration:
0: 2 1
1:
2:
3:
Move: 1 2
0: 2
1:
2: 1
3:
Move: 2 1
0:
1: 2
2: 1
3:
Move: 1 1
0:
1: 2 1
2:
3:
Final configuration:
0:
1: 2 1
2:
3:
Worked!
```

```
Number of moves: 3
$ time java-introcs Hanoi4 64 | java-introcs Hanoi4Sim 64
Worked!
Number of moves: 18433

real 0m0.354s
user 0m0.686s
sys 0m0.156s
$ time java-introcs Hanoi4 200 | java-introcs Hanoi4Sim 200
Worked!
Number of moves: 14680065

real 0m13.756s
user 0m24.376s
sys 0m4.053s
$ java-introcs Hanoi4 6 | java-introcs Hanoi4Sim 6
Worked!
Number of moves: 17
Should be at most 17
$
```

Observação 1. Note que Hanoi4Sim.java tem dois modos de execução. Quando um segundo argumento de linha de comando é dado, ele roda no modo "verborrágico", imprimindo os movimentos e a configuração das torres a cada movimento.

Observação 2. Note que, quando o número de discos é da forma $\binom{n+1}{2}$, o simulador imprime a cota superior $(n - 1)2^n + 1$. Sua solução não pode ter mais movimentos que essa cota superior.

Observação 3. Espera-se que seu Hanoi4.java tenha tempo de execução, grosso modo, não muito diferente do que se vê acima.

Observação 4. É possível resolver Hanoi4(N) ($1 \leq N \leq 50$) com os seguintes números de movimentos:

1 3 5 9 13 17 25 33 41 49 65 81 97 113 129 161 193 225 257 289 321 385 449 513 577 641 705 769 897 1025
1153 1281 1409 1537 1665 1793 2049 2305 2561 2817 3073 3329 3585 3841 4097 4609 5121 5633 6145 6657


Isto é, $U_1 \leq 1, U_2 \leq 3, \dots, U_{50} \leq 6657$. Espera-se que seu programa dê soluções não piores.

Observação 5. Em seus momentos livres :-), você poderia considerar implementar algo como AnimatedHanoi.java para 4 torres. Seu programa Hanoi4.java poderia fornecer a solução e você poderia ter uma versão gráfica de Hanoi4Sim.java, digamos Hanoi4SimAnimated.java. Seu sistema seria então composto de dois módulos independentes.

Entrega. Entregue seu programa Hanoi4.java.

-  [AnimatedHanoi.java](#) 23 outubro 2024, 17:45 PM
-  [Hanoi4Sim.java](#) 23 outubro 2024, 17:45 PM

Status de envio

Status de envio	Enviado para avaliação
Status da avaliação	Avaliado
Tempo restante	A tarefa foi enviada 1 dia 2 horas adiantado
Última modificação	sábado, 9 nov. 2024, 21:40
Envios de arquivo	<div><div> Hanoi4.java</div><div>9 novembro 2024, 21:40 PM</div></div>