

CCM0118 - Computação I (2024ii)

⌂ [Início](#) / [Meus Ambientes](#) / [2024](#) / [RUSP](#) / [CCM](#) / [0118.2024ii](#) / [Exercícios-programa](#) / [E05 Torres 3D](#)

E05 Torres 3D

Vencimento: domingo, 24 nov. 2024, 23:59

Um problema que estudamos em sala é o de determinar de quantos jeitos podemos colocar 8 rainhas em um tabuleiro de xadrez sem que nenhuma rainha ataque outra. Veja também Web Exercise 1.4.4, Web Exercise 2.3.21 e Web Exercise 2.3.22 de IntroCS:

<http://introcs.cs.princeton.edu/java/14array/>

<http://introcs.cs.princeton.edu/java/23recursion/>

Nosso problema aqui é estudar de quantas formas podemos colocar 64 torres em um tabuleiro de xadrez tridimensional 8 x 8 x 8:

https://en.wikipedia.org/wiki/Three-dimensional_chess

Em um tabuleiro 3D, há 8 níveis, cada um deles sendo um tabuleiro 8 x 8. Assim, uma posição de um tabuleiro 3D corresponde a uma linha, uma coluna e um nível (altura). Naturalmente, podemos pensar que uma posição é uma tripla (i, j, k) : i indica a linha, j indica a coluna e k indica a altura da posição no tabuleiro 3D.

No xadrez 3D, a torre pode se mover dos jeitos usuais nos tabuleiros horizontais e também pode se mover na coluna vertical que contém sua posição atual. Assim, se uma torre está na posição (i, j, k) , então ela pode se mover para a posição (i', j', k') se vale uma das condições abaixo:

1. $(i, j) = (i', j')$
2. $(i, k) = (i', k')$
3. $(j, k) = (j', k')$

A generalização de tabuleiros 8 x 8 x 8 para tabuleiros $N \times N \times N$ é óbvia. Seja $T(N)$ o número de formas de se dispor $N \times N$ torres no tabuleiro de xadrez 3D $N \times N \times N$ sem que elas se ataquem. Escreva um programa, obrigatoriamente chamado `Torres3D.java`, que determina $T(N)$ quando N é dado como argumento de linha de comando.

Exemplo. Uma disposição válida no caso $N = 4$ é codificada pela matriz

```
2 0 1 3
0 3 2 1
3 1 0 2
1 2 3 0
```

Na matriz acima, digamos a, temos $a[0][1] = 0$. Isso indica que há uma torre na posição $(0, 1, 0)$. O fato que $a[2][0] = 3$ indica que há uma torre na posição $(2, 0, 3)$. Dito de outra forma, a matriz 4 x 4 acima representa a seguinte disposição válida de 16 torres no tabuleiro 3D 4 x 4 x 4:

Nível 0:

```
. T . .
T . . .
. . T .
. . . T
```

Nível 1:

```
. . . T .
. . . T
. T . .
T . . .
```

Nível 2:

?

```
T . . .
. . T .
. . . T
. T . .
```

Nível 3:

```
. . . T
. T . .
T . . .
. . T .
```

Dicas. Para representar uma configuração de torres, use a representação matricial acima. Como deve ser cada linha de tal matriz? Como deve ser cada coluna?

Modos de execução. Seu programa Torres3D.java deve ter dois modos de execução: quando apenas N é dado na linha comando, seu programa deve imprimir $T(N)$. Se um segundo argumento de linha de comando é dado (qualquer que seja ele), seu programa deve imprimir cada uma das soluções encontradas. Siga o formato de saída no exemplo abaixo.

Exemplos de execução

```
$ java-introcs Torres3D 3
```

```
Total no: 12
```

```
$ java-introcs Torres3D 3 +
```

```
0 1 2
```

```
1 2 0
```

```
2 0 1
```

```
0 1 2
```

```
2 0 1
```

```
1 2 0
```

```
0 2 1
```

```
1 0 2
```

```
2 1 0
```

```
0 2 1
```

```
2 1 0
```

```
1 0 2
```

```
1 0 2
```

```
0 2 1
```

```
2 1 0
```

```
1 0 2
```

```
2 1 0
```

```
0 2 1
```

```
1 2 0
```

```
0 1 2
```

```
2 0 1
```

```
1 2 0
```

```
2 0 1
```

```
0 1 2
```

```
2 1 0
```

```
0 2 1
```

```
1 0 2
```

```
2 1 0
```

```
1 0 2
```

```
0 2 1
```

```
2 0 1  
0 1 2  
1 2 0  
  
2 0 1  
1 2 0  
0 1 2
```

```
Total no: 12  
$ time java-introcs Torres3D 5  
Total no: 161280
```

```
real 0m0.279s  
user 0m0.265s  
sys 0m0.071s  
$ time java-introcs Torres3D 6  
Total no: 812851200
```

```
real 2m52.647s  
user 2m51.994s  
sys 0m0.252s  
$
```

Observações. Os tempos de execução de sua implementação de Torres3D.java para N = 5 e 6 não devem ser substancialmente diferentes dos tempos acima (devem ser da mesma ordem de grandeza). Não tente determinar T(7) ou T(8) com seu programa; esses valores são muito grandes.

Status de envio

Status de envio	Enviado para avaliação
Status da avaliação	Avaliado
Tempo restante	A tarefa foi enviada 5 horas 15 minutos adiantado
Última modificação	domingo, 24 nov. 2024, 18:43
Envios de arquivo	 Torres3D.java 24 novembro 2024, 18:43 PM
Comentários sobre o envio	▶ Comentários (0)

Feedback

Nota	10,00 / 10,00
Avaliado em	domingo, 15 dez. 2024, 22:32
Avaliado por	HB Helena Baptista Reis