

ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА  
САМОПОДГОТОВКА  
ПО  
Структури от данни и програмиране  
*Привеждане на рекурсивни решения към  
решения със стек*

*email: kalin@fmi.uni-sofia.bg*

17 октомври 2017 г.

Упътване: Решете задачите с рекурсия и след това преобразувайте решението в решение със стек.

1. (\*) Да се дефинира функция за намиране на стойността на полинома на Ермит  $H_n(x)$  ( $x$  е реална променлива, а  $n$  неотрицателна цяла променлива), дефиниран по следния начин:

$$H_0(x) = 1$$

$$H_1(x) = 2x$$

$$H_n(x) = 2xH_{n-1}(x) + 2(n-1)H_{n-2}(x), n > 1,$$

за дадени  $n$  и  $x$  с използване на стек.

2. Нека е дадена абстрактна шахматна дъска с размери  $n \times n$ ,  $4 \leq n \leq 8$  и число  $k$ ,  $0 \leq k \leq n$ . Казваме, че разположени на дъската  $k$  коня образуват “валидна конфигурация”, ако никоя фигура не е поставена на поле, което се “бие” от друга фигура според съответните шахматни правила.

Да се дефинира клас `KnightConfig`, представящ “конфигуратор” на шахматни коне. Конструкторът на класа инициализира конфигуратора с числата  $n$  и  $k$ . Класът позволява “обхождането” една по една на всички валидни конфигурации за дадените параметри,

по подобие на `forward` итератор на структура от данни. Класът да притежава следните методи:

- `void KnightConfig::printCurrentConfig()`: Отпечатва текущо намерената конфигурация. Пример за отпечатана конфигурация с  $n = 5, k = 2$ :

```
- - - - -
- _ Н _ -
- - - - -
- _ _ _ Н
- - - - -
```

- `void KnightConfig::findNextConfig()`: Намира следваща конфигурация.
- `bool KnightConfig::noMoreConfigs()`: Показва дали всички възможни конфигурации са вече изчерпани.

3. Да се реши задачата за Ханойските кули с използване на стек.

Да се дефинира клас `HanoyPlayer` със следните методи:

- Конструктор с параметър, указващ броя дискове върху лявото колче за началното състояние на играта.
- Метод `bool final()`, който показва дали играта е достигнала финално състояние (т.е. всички дискове са на дясното колче).
- Метод `makeMove()`, който извършва един ход от играта.
- Метод `printBoard()`, който отпечатва текущото състояние на игровата дъска, например по следния начин:

```
2
3    1
5 *  4
```

На примера е изобразено състояние на играта, при което на лявото колче има три диска - с размери 5, 3 и 2, на средното колче няма дискове, а на дясното има два диска - с размери 4 и 1.