



Упражнения: Рекурсия, пълно изчерпване и търсене с връщане назад

Задача 1 Разходката на коня

Дадена е обобщена шахматна дъска $N \times N$ и шахматен кон, разположен в полето с координати (x_0, y_0) . Търси се такава последователност от ходове на коня в смисъла на шахматните правила за движение на тази фигура, при която всяко поле се посещава точно веднъж.

Пример:

При $n=5$ съществуват 304 различни разходки на коня. Ето две от тях:

Разходка 1	Разходка 89
1 6 15 10 21 14 9 20 5 16 19 2 7 22 11 8 13 24 17 4 25 18 3 12 23	1 10 25 16 7 20 15 8 11 24 9 2 21 6 17 14 19 4 23 12 3 22 13 18 5

Подсказки:

За решаване на задачата е необходимо да се обхождат всички клетки точно по веднъж, т.е. извършване на N^2 на брой ходове.

// Pseudo Code

Try (ход K)

while (допустим следващ ход)

 регистраме хода

 if (броя на ходовете $= N^2$) Try \leftarrow хода

 else Try \leftarrow Try (ход K+1)

 премахваме регистрацията

Край

Регистрацията на поредния ход на коня е свързана освен с маркиране на текущото поле като посетено и с добавянето му към списък от координатите на посетените полета. Според шахматните правила за всяка позиция на коня съществуват точно 8 потенциални кандидата, стига, разбира се да не извеждат коня извън границите на дъската или върху вече посетеното поле. Една от възможностите е да направите два отделни списъка на разликите, съответно за всяка от координатите. За целта може да въведете масив T, който ще съдържа координатите на относителното отместване по отношение на текущата позиция по x и y съответно. Ще считаме, че началното поле (x_0, y_0) е горният ляв ъгъл на обобщената шахматна дъска. Задачата се решава от функцията Try, която получава като параметри текущите координати на шахматния кон върху дъската, както и поредния номер на хода, който трябва да се извърши.



Задача. 2 Пътища в лабиринт

Даден е лабиринт. Задачата е да се намерят всички пътища от началната клетка, която се намира в горния ляв ъгъл (с координата - 0, 0) до изхода, маркиран със символа 'е'. Празните клетки са маркирани с тире "-", а стените със звезда "*". На първия ред се въвеждат размерите на лабиринта. На следващите редове действителния лабиринт. На изхода се извежда пътя, като последователност от символи (Up->Left->Right->Down). Редът на пътищата няма значение.

Пример:

Вход	Изход
3 3 --- -* --e	RRDD DDRR
3 5 -**-e ----- *****	DRRRRU DRRRUR

Подсказки:

Лабиринтът се представя като двумерен масив с N реда и M стълба. Започва се от горния ляв ъгъл, като може да се движите в четирите посоки - нагоре, надолу, наляво и надясно. Проходимите клетки са отбелязани с тире "-", а непроходимите - със звезда "-*".

Да разгледаме първия примерен вход:

-(0, 0)	-	-
-	*	-
-	-	e

RRDD

-(0, 0)	-	-
-	*	-
-	-	e

DDRR

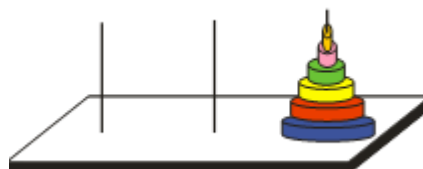
Идеята за решаване на тази задача е рекурсия, като се използва backtracking. Нека сме в клетка с координати (X,Y). Стъпки на рекурсията:

- търси_изход(X,Y+1)
- търси_изход(X+1,Y)
- търси_изход(X,Y-1)
- търси_изход(X-1,Y)



Задача. 3 Ханойски кули

Дадени са три стълба. На първия са поставени n диска с различен диаметър, наредени един върху друг от най-големия към най-малкия диск. Задачата е да се преместят всички дискове на третия стълб, като се запази подредбата им и при разместванията се спазва правилото винаги да се поставя по-малък диск върху по-голям.



Пример:

Вход	Изход
3	1-->3 1-->2 3-->2 1-->3 2-->1 2-->3 1-->3

Подсказки:

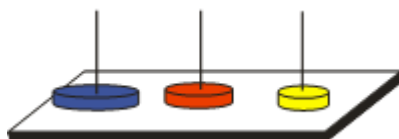
Нека разгледаме задачата при $n=3$.



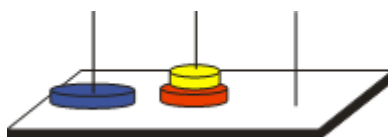
1 стъпка:



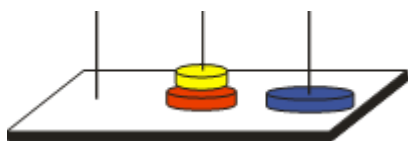
2 стъпка:



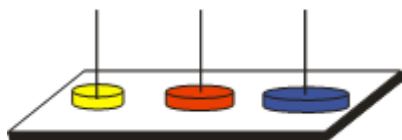
3 стъпка:



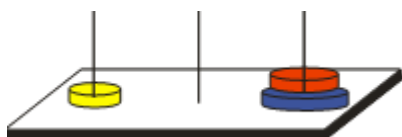
4 стъпка:



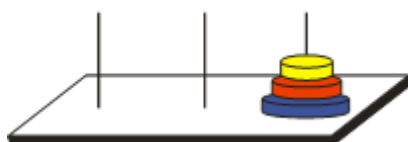
5 стъпка:



6 стъпка:



7 стъпка:



Броят на стъпките за решаване на задачата е $2^n - 1$. При $n=3$ получаваме $2^3 - 1 = 8 - 1 = 7$.

Задачата за n диска се свежда до задача за $n-1$ диска по следния начин:

- преместваме горните $(n-1)$ диска на стълб 2, като използваме стълб 3 за помощен
- преместваме n -тия диск от стълб 1 на стълб 3
- преместваме $(n-1)$ -те диска от стълб 2 на стълб 3, като използваме стълб 1 за помощен

Когато дисковете са 4 на брой, целта ни е да поставим най-големия диск на мястото му на третия стълб, после разиграването на останалите 3 диска ще се сведе до вече решавана задача. Когато дисковете станат 5, отново търсим начин да поставим най-големия на третия стълб, и после следва решаваната задача с 4 диска и т.н. По този начин общата задача се разбива на по-малки задачи от същия тип, което води до рекурсивно решение. За рекурсивната функция са ви необходими 4 параметъра: I - броя на дисковете, II - от кой стълб взимаме, III - кой стълб е помощен и IV - на кой стълб поставяме диска.