Antonescu Cristina Andreea

Proiect realizat cu tehnica de programare “Backtracking”

**DOCUMENTATIE**

La o masa rotunda trebuie asezati n cavaleri care trebuie asezati pe n scaune, numerotate de la 1 la n. Nu toti cavalerii sunt prieteni intre ei - ei trebuie asezati astfel incat pe doua scaune alaturate sa stea doar cavaleri prieteni. Se cer toate solutiile de asezare a cavalerilor.

**Date de intrare**: fișierul date.in conține:

- pe prima linie numarul N, cu semnificația din enunț;

-pe următoarele linii perechi de forma i j cu semnificația că cei doi cavaleri, i si j, sunt prieteni.

**Date de ieșire**: fișierul date.out conține pe fiecare linie câte o soluție; o soluție este de forma a N numere, separate prin cate un spatiu, reprezentand cavalerii, in ordinea in care sunt asezati pe scaune.

**Restricții și precizări:**

N∈[2,102]

Despre cod:

Deoarece aceasta problema se rezolva cu ajutorul tehnicii de programare “backtracking”, vom utiliza in programul nostru variabile globale. Astfel, primul pas pe care il voi face in rezolvarea acestei probleme va fi de a crea fisierele cerute si de a initializa variabilele cu care voi lucra global. “n” este numarul de cavaleri ce se vor aseza la masa rotunda, “a[105][105]” este matricea adiacenta, “x[105]” reprezinta vectorul ce va contine solutiile problemei, “y[105]” este vectorul de aparitii, iar “nrsol” identifica numarul de solutii.

Mai departe am realizat o functie numita “citire” ce face exact ce ii sugereaza si numele, citeste din fisierul de intrare “date.in” numarul n de cavaleri si acele perechi de cavaleri prieteni (i si j) si la final initiaizeaza cu 1 atat a[i][j] cat si a[j][i].

Functia “afisare” va mari “nrsol” cu o unitate de fiecare data cand se trece prin ea si va afisa cate o solutie pe cate o linie din vectorul de solutii “x[i]” in fisierul de iesire “date.out”.

Urmatoarea functie pe care am implementat-o se numeste “valid”. Ea are rolul de a verifica daca o solutie este partiala. Aici se stabilesc conditiile in care relatiile de prietenie se pot infaptui/exista intre cavaleri: Daca o valoare deja aparuta este diferita de 0- returnam 0, daca k e mai mare ca 1 si nu am nimic de la elemental anterior pana la cel current- returnam 0 si daca de la elemtul current pana la elemental 1 al matricei am 0- returnam 0. Altfel returnam 1.

In sfarsit ajungem la functia de backtracking pe care am numit-o sugestiv “bkt”. Daca am generat toate cele n elemente, apelam “afisare”- deci ne intoarcem la pasul anterior, altfel iau toate elementele de la 1 la n si pun valoarea curenta pe pozitia k. Daca e o valoare buna, marcam cu 1 vectorul de aparitii, apelez bkt(k+1) si, dupa ce executa din nou functia recursiva, ne intoarcem in apel si demarcam vectorul de aparitii.

In functia “main”, finalizam crearea fisierelor de intrare, apelam “citire”, iar pentru a nu ne aparea o solutie de 2 ori, initializam vectorul de solutii cu 1 si apoi l-am marcat, dupa care am apelat “bkt” de la a 2-a pozitie. La final am inchis fisierele.

Complexitate:

Pe baza complexitatii datelor de iesire, putem spune ca acest algoritm are complexitatea maxima 𝒪((𝑛-1)!).

Exemple:

|  |  |  |
| --- | --- | --- |
| Date.in | Date.out | Precizari: |
| 4  1 2  2 3  3 4  4 1 | Solutia 1  1 2 3 4 1  Solutia 2  1 4 3 2 1 | Soldatii sunt prieteni crescator |
| 5  1 2  1 3  1 4  1 5  2 3  2 4  2 5  3 4  3 5  4 5 | Solutia 1  1 2 3 4 5 1  Solutia 2  1 2 3 5 4 1  Solutia 3  1 2 4 3 5 1  Solutia 4  1 2 4 5 3 1  Solutia 5  1 2 5 3 4 1  Solutia 6  1 2 5 4 3 1  Solutia 7  1 3 2 4 5 1  Solutia 8  1 3 2 5 4 1  Solutia 9  1 3 4 2 5 1  Solutia 10  1 3 4 5 2 1  Solutia 11  1 3 5 2 4 1  Solutia 12  1 3 5 4 2 1  Solutia 13  1 4 2 3 5 1  Solutia 14  1 4 2 5 3 1  Solutia 15  1 4 3 2 5 1  Solutia 16  1 4 3 5 2 1  Solutia 17  1 4 5 2 3 1  Solutia 18  1 4 5 3 2 1  Solutia 19  1 5 2 3 4 1  Solutia 20  1 5 2 4 3 1  Solutia 21  1 5 3 2 4 1  Solutia 22  1 5 3 4 2 1  Solutia 23  1 5 4 2 3 1  Solutia 24  1 5 4 3 2 1 | Toti soldatii sunt prieteni |
| 7  1 2  4 3  5 2  6 2  7 2  1 7  1 6  3 7  5 4 | Solutia 1  1 6 2 5 4 3 7 1  Solutia 2  1 7 3 4 5 2 6 1 | Soldatii sunt prieteni aleator |