

Rezolvarea problemelor de căutare



Obiective

Formularea problemelor ca probleme de căutare și identificarea modalităților de rezolvare a lor.
Specificarea, proiectarea și implementarea metodelor de căutare.



Aspecte teoretice

Rezolvarea problemelor ca proces de căutare

Tipuri de probleme de căutare.

Modalități de rezolvare a problemelor de căutare → construirea progresivă a soluției:

- Stabilirea componentelor problemei
 - Stare inițială
 - Stare finală
 - Operatori (funcții succesor)
 - Soluție
- Definirea spațiului de căutare
- Stabilirea strategiei de identificare a soluției în spațiul de căutare



Termen de predare

Laborator 2

Cerințe



Specificați, proiectați și implementați o aplicație care să rezolve una dintre următoarele probleme. Aplicația trebuie să permită:

- Încărcarea datelor problemei
- Alegerea și parametrizarea metodei de rezolvare a problemei
- Afișarea soluției identificate
- Precizarea performanțelor metodei de rezolvare alese

Fiecare dintre probleme trebuie rezolvată cu cele 2 tehnici precizate (dar se pot adăuga și tehnici noi) – o tehnică neinformată (breadth-first search (BFS) sau depth-first search (DFS)) și o tehnică de căutare locală (Greedy).

Aplicația trebuie să respecte specificațiile privind datele de intrare și datele de ieșire.

Aplicația va fi testată folosind date de dificultăți diferite (fiecare test validat având asociat un anumit punctaj).

De asemenea, pentru fiecare problemă există exemple de date de test.

Codul aferent aplicației poate fi dezvoltat 100% de către student sau poate fi bazat pe exploatarea unor biblioteci specifice.

Alg	Code 100%					Code&tools					Min for valid
	Code	Easy Test	Medium Test	Hard Test	Total	Code	Easy Test	Medium Test	Hard Test	Total	
DFS/BFS	100	50	80	100	330	50	10	50	80	190	110
Greedy	100	50	80	100	330	50	10	50	80	190	110
Total	200	100	160	200	660	100	20	100	160	380	

1. Jocul Sudoku

Dându-se un joc Sudoku (un puzzle logic reprezentat pe o tablă cu $n \times n$ căsuțe; unele căsuțe conțin deja câte un număr, altele trebuie completate cu alte numere din $\{1, 2, \dots, n\}$ astfel încât pe fiecare linie, fiecare coloană și fiecare pătrățel cu latura egală cu \sqrt{n} pătrățele să conțină doar numere diferite), să se determine o modalitate corectă de completare a căsuțelor libere.

3			2
	1	4	
1	2		4
	3	2	1

	2		6		8			5
5	8				9	7		
		7		4			2	8
3	7		4		1	5		
6				8				5
		8			2		1	3
8		6		2		1		
		9	8				3	6
7			3		6		9	

Figura 1 a) joc Sudoku cu 4x4 căsuțe; b) joc Sudoku cu 9x9 căsuțe

2. Jocul criptaritmeticii

Dezvoltați un algoritm care să rezolve eficient oricare dintre problemele de criptaritmetica prezentate în Figura știind că:

- Fiecare literă reprezintă o cifră zecimală;
- Rezultatul operației aritmetice trebuie să fie corect atunci când literele sunt înlocuite cu cifre;
- Numerele nu pot începe cu cifra 0;
- Fiecare problemă poate avea o singură soluție.

$$\begin{array}{r} \text{SEND} + \\ \text{MORE} = \\ \hline \text{MONEY} \end{array}$$

Figura 2 Problema de criptaritmetică

3. Problema comisului voiajor

Dându-se o hartă cu mai multe orașe și distanțele dintre ele, să se găsească cel mai scurt drum care trece o singură dată prin toate orașele.

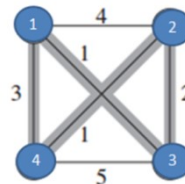


Figura 3 Problema comisului voiajor cu 4 orașe

4. Problema rucsacului

Dându-se un rucsac cu o anumită capacitate și un număr de obiecte (fiecare obiect având o anumită greutate și un anumit cost), să se umple cât mai bine rucsacul cu obiecte cât mai valoroase.

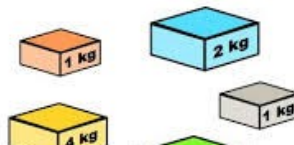


Figura 4 Problema rucsacului cu 5 obiecte