

SEMINAR 13+14:

S13: DIVIDE & CONQUER + BACKTRACKING

S14: METODA GREEDY + PROGRAMARE DINAMICĂ

1. DIVIDE & CONQUER

Pas 1: **Divide**

- se împarte problema în probleme mai mici (de același structură)
- împărțirea problemei în două sau mai multe probleme disjuncte care se poate rezolva folosind același algoritm

Pas 2: **Conquer** – se rezolvă subproblemele recursiv

Pas 3: **Combine** – combinarea rezultatelor

- 1.1. Verificați dacă într-o listă dată de la tastatură există un număr impar.
- 1.2. Verificați dacă toate numerele dintr-o listă dată sunt numere negative.
- 1.3. Calculați numărul de apariții al unui număr dat într-o listă.
- 1.4. Calculați suma numerelor pare dintr-o listă dată.
- 1.5. Calculați cel mai mare divizor comun al numerelor dintr-o listă dată.
- 1.6. Găsiți cel mai mare număr prim dintr-o listă dată.
- 1.7. *Găsiți al k-lea cel mai mare număr dintr-o listă dată.

2. BACKTRACKING

Formalizare soluție:

- Definim spațiul de căutare $S = S_1 \times S_2 \times \dots \times S_n$
 - S_i = valorile pe care lua componenta i a soluției
- Vectorul x va reprezenta o soluție la modul general
 - $x = (x_1, x_2, \dots, x_k)$, $x_i \in S_i$, reprezintă o **soluție candidat** = configurație parțială care poate (sau nu) conduce la un rezultat, k = numărul de componente din candidatul curent, $k < n$
 - **consistent** - o funcție care verifică dacă o soluție candidat poate conduce la un rezultat
 - **soluție** - o funcție care verifică dacă o soluție candidat este soluție pentru problemă

Se cere formalizarea următoarelor probleme de backtracking, și scrierea unei variante recursive și a unei variante iterative pentru rezolvarea acestora.

1. Se citește un cuvânt format din maxim 10 litere mici distincte. Afișați anagramele cuvântului citit:
 - a) toate, în ordine lexicografică
 - b) anagramele care au proprietatea că nu conțin două vocale alăturate și nici două consoane alăturate (i.e. vocalele și consoanele trebuie să alterneze)
2. Se citește un număr natural n . Generați și afișați toate combinațiile de câte $2 \cdot n + 1$ cifre binare care nu au 2 cifre de 1 alăturate.
3. Se citește un număr natural n cu cel mult 9 cifre. Afișați toate modalitățile de a-l scrie pe n ca produs de divizori proprii distincți ai lui s .
4. Se citesc două numere naturale n și s ($n \leq 10$, $s \leq 20$). Afișați în ordine crescătoare toate numerele cu n cifre care au suma cifrelor egală cu s și în care oricare 2 cifre alăturate au paritate diferită.
5. Se citește un număr natural n . Afișați permutările mulțimii $1, 2, \dots, n$ în care elementele pare sunt

puncte fixe (se află pe poziții egale cu valoarea lor)

6. Să se afișeze toate sublistele crescătoare de lungime >1 pentru o listă de n numere dată.

3. METODA GREEDY

1. Se dau o serie de activități care trebuie realizate de o persoană într-o zi dată. Fiecare activitate este caracterizată de ora de început și ora de sfârșit a ei. Se cere determinarea numărului maxim de activități care pot fi realizate, presupunând că persoana nu poate lucra la două activități în același timp.

Exemple:

Activitate	1	2	3
Ora de început	2	5	7
Ora de finalizare	6	8	12

Soluție:

Activitate	1	2	3	4	5	6	7	8
Ora de început	9	12	8	10	16	14	20	19
Ora de finalizare	11	13	10	12	18	16	22	21

Soluție:

2. Se cere ocuparea optimă a unui camion care poate transporta o greutate maximă G cu n obiecte, fiecare obiect având asociată greutatea g_i și un profit asociat, p_i . Din fiecare obiect se poate lua o fracțiune.

Exemple:

Obiect	1	2	3
Greutate	10	20	30
Profit	60	100	120

$G = 50$

Soluție

Item			Fracțiune	Profit
Nr	Greutate	Profit		
			Profit total	

Obiect	1	2	3	4	5
Greutate	5	4	4	8	10
Profit	10	20	10	10	22

$G = 20$

Soluție

Item			Fracțiune	Profit
Nr	Greutate	Profit		

			Profit total	

4. PROGRAMARE DINAMICĂ

1. *Se dă o listă de numere. Să se determine lungimea sublistei crescătoare de lungime maximă.

Exemplu:

Index	0	1	2	3	4	5
L	1	-2	3	2	4	4
lungimi_max						
indici_max						

2. Într-un rucsac se poate transporta o greutate maximă **G** și există **n** obiecte, fiecare având greutatea **g_i** și un profit asociat, **p_i**. Obiectele nu pot fi fracționate. Să se determine profitul maxim care poate fi obținut prin obiectele transportate în rucsac.

Exemplu:

N = 4

G = 10

Obiect	1	2	3	4
Greutate	2	3	6	4
Profit	20	40	50	45

		Greutate disponibilă										
		0	1	2	3	4	5	6	7	8	9	10
Obiect	1											
	2											
	3											
	4											

Completați tabelul pentru:

N = 3

G = 6

Obiect	1	2	3
Greutate	1	2	3
Profit	2	15	40

3. Se consideră un triunghi de numere naturale a_{ij} cu n linii. Pornind de la numărul din linia 1, mergând în jos până la linia n , să se determine **o selecție de elemente** astfel încât suma elementelor să fie maximă. Trecerea la linia următoare se poate face doar mergând în jos, direct sau pe diagonală (la dreapta).

	1	2	3	4
--	---	---	---	---

1	5			
2	4	2		
3	5	4	3	
4	4	3	2	5