



Laborator 21

În **REPORT.txt** adăugați output-ul versiunilor finale ale programelor. Dacă o parte din program nu e implementată, nu funcționează, face ca programul să dea seg fault atunci puteți comenta unele linii din main și să folosiți aceea afișare.

Exerciții

Dintr-un element $a_{i,j}$ al unei matrice $A_{n,n}$ se poate ajunge în elementele $a_{i+1,j}$, $a_{i+1,j+1}$, $a_{i+1,j-1}$. Știind că fiecare element al matricei reține un număr natural, se cere un drum care îndeplinește condițiile problemei și unește un element de pe linia 1 cu unul pe linia n , astfel încât suma numerelor reținute de elementele pe unde trece drumul să fie maximă.

1. **(greedy.c)** Să se rezolve problema folosind metoda de programare *greedy*.
2. **(backtracking.c)** Să se rezolve problema folosind metoda de programare *backtracking*.
3. **(dp.c)** Să se rezolve problema folosind metoda *programare dinamică*.
4. Pentru fiecare dintre metodele de rezolvare de la exercițiile 1-3, să se măsoare timpii de execuție și să se optimizeze cât se poate de mult. Rezultatele testelor vor fi salvate în fișierul **timpi.csv**.

Exercițiile de la 1 la 4 sunt obligatorii. Conceptele explorate sunt esențiale pentru obținerea notei **minime** de promovare.

Exemplu

Input: $n = 3$; $A = \begin{pmatrix} 3 & 1 & 3 \\ 4 & 1 & 2 \\ 3 & 7 & 1 \end{pmatrix}$

Output: Drumul este: $a_{1,1}, a_{2,1}, a_{3,2}$, iar suma este $3 + 4 + 7 = 14$