Antonescu Cristina Andreea

Proiect realizat cu tehnica de programare “Programare Dinamica”

**DOCUMENTATIE**

Alice s-a pierdut din nou în labirint. Labirintul este de forma unui triunghi dreptunghic împărțit în camere organizate pe linii și coloane. Pe prima linie este o singură cameră, pe a doua sunt două camere, etc. Pe ultima linie sunt n camere. Din fiecare cameră se poate merge în camerele situate pe linia următoare și pe aceeași coloană, sau pe coloane cu 1 mai mari sau mai mici decât coloana curentă, dar fără să se părăsească labirintul. Astfel din camera (i,j) se poate merge în camerele (i+1,j-1), (i+1,j) și (i+1,j+1), dacă acestea există. Alice se află în camera de pe prima linia și de pe prima coloana (1,1).

Ca să iasă din labirint, Alice trebuie să ajungă într-o cameră de pe ultima linie. Calculați și afișați numărul de trasee pe care poate Alice să iasă din labirint.

Despre cod:

Pentru acest cod am ales sa lucrez doar in main. Pentru inceput am initializat variabilele pe care le voi folosi: matricea A si constanta sol = 0 (numarul de trasee) le-am atribuit dimensiunea unui long, iar pentru numarul de linii al matricei (n) am ales tipul int. Am setat prima pozitie din matrice cu 1 si apoi am inceput sa pun conditii. Am tratat cateva cazuri ‘exceptionale’: **daca** j = 1, atunci pentru pozitia A[i][j] vom avea doar 2 cazuri: Alice poate veni doar de pe pozitiile A[i-1][j] si A[i-1][j+1]; **daca** j = i – 1, atunci pentru pozitia A[i][j] vom avea tot doar 2 cazuri: A[i -1][j - 1] si A[i - 1][j]; **daca** j = I, atunci pentru pozitia A[i][j] va exista doar un caz: A[I - 1][j - 1]. Daca niciunul din aceste cazuri nu este indeplinit, atunci pentru pozitia A[i][j] vom avea 3 cazuri: A[I - 1][j -1], A[I - 1][j], A[I - 1][j + 1]. La final, dupa toate aceste operatii, solutia pe care trebuie sa o afisam se va afla pe pozitia A[n][i] si o vom atribui lui sol.

Complexitate:

Compexitatea algoritmului este aproximativ egala cu 𝒪(𝑛2) datorita celor 2 for-uri.

Exemple:

|  |  |  |  |
| --- | --- | --- | --- |
| Nr.crt. | Date de intrare-  Nr. Linii matrice: | Date de iesire  Nr. trasee: | Explicatie: |
| 1 | n = 3  *1*  *2 3*  *4 5 6* | 5 | 1-2-4  1-2-5  1-3-4  1-3-5  1-3-6 |
| 2 | n = 4  *1*  *2 3*  *4 5 6*  *7 8 9 10* | 13 | 1-2-4-7 1-3-4-7  1-2-4-8 1-3-5-8  1-2-5-8 1-3-5-9  1-2-5-7 1-3-5-7  1-2-5-9 1-3-6-8  1-3-4-8 1-3-6-9  1-3-6-10 |
| 3 | n = 5  *1*  *2 3*  *4 5 6*  *7 8 9 10*  *11 12 13 14 15* | 35 | 1-2-4-7-11 1-3-4-8-13  1-2-4-7-12 1-3-5-7-11  1-2-4-8-11 1-3-5-7-12  1-2-4-8-12 1-3-5-8-11  1-2-4-8-13 1-3-5-8-12  1-2-5-7-11 1-3-5-8-13  1-2-5-7-12 1-3-5-9-12  1-2-5-8-11 1-3-5-9-13  1-2-5-8-12 1-3-5-9-14  1-2-5-8-13 1-3-6-8-11  1-2-5-9-12 1-3-6-8-12  1-2-5-9-13 1-3-6-8-13  1-2-5-9-14 1-3-6-9-13  1-3-4-7-11 1-3-6-9-14  1-3-4-7-12 1-3-6-10-13  1-3-4-8-11 1-3-6-10-14  1-3-4-8-12 1-3-6-10-15 |