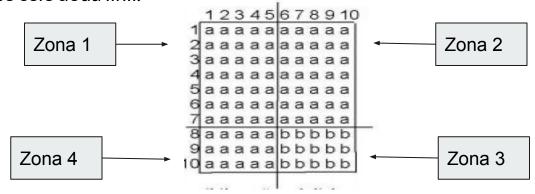
Problema 9 - Joc Documentatie

Bleoanca Diana Iulia

Cerinta problemei

Se considera un joculet cu doi jucatori. Unul este utilizatorul, celalalt computer-ul. Se deseneaza o matrice de 10 pe 10 linii, initial fiecare unitate avand valoarea 'a'. Utilizatorul trebuie sa traga doua linii, una orizontala, una verticala, pentru a imparti tabla in 4 zone, urmand sa aleaga o zona in care sa modifice cu valori 'b'. Punand cat mai putine intrebari (interogheaza ce se afla la o anumita pozitie), calculatorul trebuie sa-si dea seama unde sunt trase cele doua linii.



Mod de abordare

Algoritmul implementat foloseste logica Cautarii binare pentru a scoate un timp cat mai bun de cautare.

Ideea algoritmului incepe prin a interoga fiecare colt pana il gaseste pe cel potrivit(in care primiste ca raspuns 'b'), urmand sa puna intrebari pe margini(pe linia orizontala si pe cea verticala), la pozitii aflate prin injumatatire.

Mod implementare

Pentru implementarea cerintei au fost create trei module cu extensie py:

- 1. functii.py modul in care se regasesc toate functiile necesare rezolvarii jocului
- **2. generator.py** modul in care se definesc functii pentru generarea random de input-uri posibile;
- **3.** *main.py* modul principal in care se apeleaza functiile definite. In acest modul se realizeaza scrierea in fisiere.

Modulul functii.py

Modul de functii - Clase necesare rezolvarii jocului

- Clasa Matrice
 - Clasa ce instantiaza matricea jocului(cea creata de utilizator)
 - ☐ Operanzi din clasa, instantiati in __init__:
 - Linie1 pozitia la care linia orizontala este trasata
 - Linie2 pozitia la care linia verticala este trasata
 - Zona zona unde s-a ales a fi completat cu 'b'
 - Module ale clasei:
- Functia find(self, pozitie1, pozitie2)
 - > Parametrii: Cei doi indici unde se face interogarea.
 - > Functia returneaza caracterul ce se afla pe pozitia specificata

Clasa subMatrice

- Clasa ce ajuta la parcurgerea algoritmului. Are scopul de a memora aria suprafesei necucerite la fiecare moment
- ☐ Operanzi din clasa, instantiati in __init__:
 - iMin- valoarea minima a liniilor
 - iMax valoarea maxima a liniilor
 - ¡Min valoarea minima a coloanelor
 - jMax valoarea maxima a coloanelor
- ☐ Module ale clasei:
- Functia colt(self, nrColt)
 - Parametrii: Numarul coltului din matrice la care se cer pozitiile
 - Functia returneaza pozitiile coltului specificat

- Functiile mijloc_linie/mijloc_coloana(self)
 - Functiile au rolul de returna indicele din mijlocul liniei/coloanei submatricii

- Functia injumatatire_matrice_linie/injumatatire_matrice_coloana(self, pozitia, zona, raspuns)
 - Parametrii: pozitita reprezinta indicele la care s-a pus ultima interogare, zona reprezinta numarul cadranului, iar raspunsul este caracterul gasit pe ultima pozitie interogata.
 - Functiile au rolul de a micsoare suprafata subMatricei, adica aria zonei nedescoperite pana in momentul de fata

Clasa Calculator

- ☐ Clasa ce instantiaza logica jucatorului calculator in scopul aflarii raspunsului corect
- ☐ Operanzi din clasa, instantiati in __init__:
 - setIntrebari lista in cate se depoziteaza intrebarile ce au fost deja puse
 - numarIntrebari intreg ce retine numarul de intrebari puse
 - puncteNImerite intreg ce retine de cate ori intrebarea a fost corecta
 - Sub_matrice instanta a clasei subMatrice, initializata cu capetele matricei de inceput
 - Colt lista formata din 3 intregi zona buna si cele 2 pozitii ale coltului acestei zone
 - Raspuns lista formata din 2 elemente contine resunsul, initial fiind 0, 0
- ☐ Module ale clasei:

- Functia fabrica_intrebari(self)
 - Functiile au rolul de returna cei 2 indici unde urmeaza a se face interogarea. Aceasta creaza intrebarea pe baza a nivelului la care se afla(cate din cele 2 raspunsuri stie in acel moment)

- Functia adaugareRaspuns(self, pozitie1, pozitie2, raspuns)
 - ➤ Parametrii: pozitie1 si pozitie2 reprezinta indicii la care s-a pus ultima interogare, iar raspunsul este caracterul gasit pe ultima pozitie interogata.
 - Functia are rolul de a memora ultima intrebare, iar pe baza acesteia si a raspunsului dat de utilizator, sa iti reconstruiasca sub_matricea ce il ajuta in formularea pozitiilor urmatoare.

Modulul generator.py

Modul pentru generarea valorilor random - Testarea algoritmului in vederea rezolvarii jocului

Modulul implementeaza generatoare random pentru urmatoarele:

- ☐ Pozitia la care se traseaza linia orizontala
- Pozitia la care se traseaza linia verticala
- Una din cele 4 zone, reprezentand suprafata in care se va completa cu 'b'

Fisiere test

Programul creeaza cinci fisiere test in care se afiseaza output-ul dupa o serie de operatii pe valori generate random.

Aceste fisiere se creeaza dupa prima rulare a programului, urmand a fi actualizate de la fiecare rulare, deoarece valorile se schimba de fiecare data.

Interfata grafica

Daca doriti sa incercati, am creat si o interfata grafica al acestui joculet. Aceasta se deschide din modulul "executor.py" si permite doua moduri de joc:

- You vs PC trebuie sa oferi ca input pozitiile liniilor trasate si zona. In continuare, puteti urmari acelasi algoritm prezentat mai sus, insa pas cu pas pe o matrice.
 Computer-ul va dezvalui locatiile unde se pune interogarea la fiecare moment.
- PC vs You Calculatorul va alege random unde sunt trasate liniile si unde se afla zona.
 Tu trebuie sa dezvalui cat mai putine locatii din matrice pentru a afla raspunsul.
 Pentru asta, trebuie completata zona de mai jos, iar jocul iti va spune daca e bine.

Observatie: !!! Dupa orice introducere in interfata, este necesar a se da enter.