Aufgabe 1: Schiebeparkplatz

Team-ID: 00921

Team: Ctrl + Intelligence

Bearbeiter/-innen dieser Aufgabe:
Thomas Grunenberg

21. November 2021

Inhaltsverzeichnis

Lösungsidee

Als Erstes muss bei einem Auto überprüft werden, ob dieses überhaupt von einem anderen Auto blockiert wird. Sollte ein Auto von einem anderen Blockiert werden muss dieses verschoben werden, entweder nach links oder nach rechts. Wenn das Auto beim nach links oder rechtsfahren blockiert wird muss das blockierende Auto ebenfalls in die Richtung des Autos verschoben werden, welches Verschoben werden muss. Es gibt keine Lösung in der um, ein Auto das Ausparken zu ermöglichen, ein Auto nach links und ein anderes nach rechts verschiben werden muss, alle Autos müssen nur in eine Richtung verschoben werden. Um herauszufinden welches Effektiver ist, also nach links oder rechts Fahren, führt man beides aus und vergleicht danach die Nummer an Zügen, die für die Lösung für eine Verschiebung nach links und nach rechts benötigt werden.

Umsetzung

Zuerst lasse ich die Input Textdatei auslesen und in ein 2.dimensionales Array einspeichern, in welchem die 1. Dimension die geparkten Autos mit Bezeichnung A-x stehen und in der 2. Dimension die quer geparkten Autos. Danach geht man für alle Autos der 1. Dimension durch, ob sie beim ausparken von einem anderen Fahrzeug Blockiert werden oder nicht, sollte dem nicht der Fall sein gibt man eine Leere Ausgabe, da kein Fahrzeug verschoben werden musste. Sollte das Fahzeug allerdings blockiert werden, versucht man das Fahrzeug durch Bewegungen des Querstehenden Fahrzeugs nach links und nach rechts zu befreien. Sollte das Auto beim nach links bzw. Beim nach rechts Fahren blockiert werden, wird durch einen rekursiven Aufruf das blockierende Auto in die selbe richtung verschoben. Danach wird verglichen welche Richtung weniger Bewegungen erfordert hat, wenn beide gleich viele Bewegungen brauchten, wird verglichen, welche Richtung weniger Fahrzeuge verschieben musste. Danach wird die Ausgabe betätigt welche als effektiver Erkannt wurde. Sollte es keine Lösung für einen Punkt geben wird ebenfalls eine Entsprechende Ausgabe betätigt.

Beispiele

Um das Programm auszuführen, muss die Batch-Datei „Aufgabe1Starten“ gestartet werden, daraufhin erhält man einen Prompt und muss nun den Dateipfad der zu testenden Datei eingeben (der Name der Datei muss mit Endung eingegeben werden, befindet diese sich im gleichen Ordner wie die Batchdatei, kann man auch nur den Namen eingeben und kann sich den Pfad sparen).

Wenn es erfolgreich war, wird für jedes Fahrzeug der optimale Weg ausgegeben um es Ausparken zu können.

Beispiel 1:

```
Loesen eines Schiebeparkplatzes
Bitte geben sie den Dateinamen an: parkplatz0.txt
A:
B:
C: H 1  right
D: H 1  left
E:
F: H 1  left I 2  left
G: I 1  left
```

Dies ist die Ausgabe für parkplatz0.txt. Man sieht das die Ausgaben wie in der Aufgabenstellung so gewählt sind, dass das zuerst zu bewegende Auto auch zuerst in der Ausgabe steht.

Beispiel 2:

```
Loesen eines Schiebeparkplatzes
Bitte geben sie den Dateinamen an: parkplatz3.txt
A:
B: O 1  right
C: O 1  left
D:
E: P 1  right
F: P 1  left
G:
H:
I: Q 2  left
J: Q 1  left
K: Q 2  left R 2  left
L: Q 1  left R 1  left
M: Q 2  left R 2  left S 2  left
N: Q 1  left R 1  left S 1  left
Drücken Sie eine beliebige Taste . . .
```

Das ist die Ausgabe für parkplatz3.txt. Hier sieht man, dass das Programm keine Schwierigkeiten damit hat auch mehrere Autos in eine Richtung zu verschieben.

Quellcode

```
public static void loesen(char[][] pl) {                //Geht alle stellen im Array von A-x durch und
                                                        //überprüft ob
    for (int a = 0; a < i; a++) {                       //das Auto raussfahren kann oder blockiert wird
      if (pl[a][1] ==0) {
        System.out.println(pl[a][0] + ":");
      } else {
        setAl(pl);
        System.out.println(pl[a][0] + ":"+ Bewegen(a,pl));
      }
    }
  }
  public static String Bewegen(int pos, char[][] pl){   //Setzt die Globalen Zählvariablen für bewegeRechts
                                                        //und
    R=0;                                                //bewegeLinks=0 um nach deren aufruf zu vergleichen,
                                                        //welche
    L=0;                                                //von den Beiden weniger Züge, bzw weniger Autos im
                                                        //falle
    rUsedCars=0;                                        //eines Gleichstandes verwendet hat und gibt die
    lUsedCars=0;                                        //Effizientere bzw überhaupt lösbare Version zurück
    setAl(pl);
    String l=bewegeLinks(pos);
    setAl(pl);
    String r=bewegeRechts(pos);
    setAl(pl);
    String y="";
    if(R<L) {y=r;}
    else if(L<R){y=l;}
    else if(L==R && rUsedCars<lUsedCars){y=r;}
    else if(L==R && lUsedCars<rUsedCars){y=l;}
    if(R>100 && L>100){
      y="Error cant solve";
    }
    return y;
  }
```

```
public static String bewegeRechts(int pos){            //bewegeRechts und bewegeLinks tun fast dasselbe
   String y="";                                         //sie versuchen das Auto so weit wie möglich nach
   int move=0;                                          //rechts bzw links zu verschieben um die Gegebene
                                                        //position
   char a=al[pos][1];                                   //frei zu machen und eben danach die gemachten
                                                        //Bewegungen
   try{                                                 //als String in der richtigen Formatierung zurück
     while(al[pos][1]!=0){                              //für den fall dass das Auto von einem Anderen
                                                        //Blockiert
       if(al[pos+1][1]==0){                             //kommt es zum rekursiven Aufruf
         R++;                                           //sollte es in die Gegebene Richtung keine Lösung
                                                        //geben weil
         move++;                                        //die Grenzen des Arrays überschritten werden, wird R
                                                        //bzw L
         shiftRight(pos);                               //gleich Maximum gesetzt damit es in Bewegung nicht
                                                        //ausgewählt werden kann
       }else if(al[pos+1][1]==al[pos][1] && al[pos+2][1]==0){
         R++;
         move++;
         shiftRight(pos);
       }else if(al[pos+1][1]!=0 && al[pos+1][1]!=al[pos][1]){
         rUsedCars++;
         y=bewegeRechts(pos+2);
       }else if(al[pos+1][1]==al[pos][1] && al[pos+2][1]!=0){
         rUsedCars++;
         y=bewegeRechts(pos+2);
       }if(R>1000){return "N";}
     }
     y=y+" "+a+" "+move+" "+" right";
   }catch(ArrayIndexOutOfBoundsException e){
     R=2000;
     al[pos][1]=0;
     return "N";
   }
   return y;
 }
 public static String bewegeLinks(int pos){
   String y="";
   int move=0;
   char a=al[pos][1];
   try{
     while(al[pos][1]!=0){
       if(al[pos-1][1]==0){
         L++;
         move++;
         shiftLeft(pos);
       }else if(al[pos-1][1]==al[pos][1] && al[pos-2][1]==0){
         L++;
         move++;
         shiftLeft(pos);
       }else if(al[pos-1][1]!=0 && al[pos-1][1]!=al[pos][1]){
         lUsedCars++;
         y=bewegeLinks(pos-2);
       }else if(al[pos-1][1]==al[pos][1] && al[pos-2][1]!=0){
         lUsedCars++;
         y=bewegeLinks(pos-2);
       }if(L>1000){return "N";}
     }
     y=y+" "+a+" "+move+" "+" left";
   }catch(ArrayIndexOutOfBoundsException e){
     L=2000;
     al[pos][1]=0;
     return "N";
   }
   return y;
 }
```

```
public static void shiftLeft(int position){            //Bewegt das Auto im Array um 1 nach links
    if(al[position][1]==al[position-1][1]){
      al[position-2][1]=al[position][1];
      al[position][1]=0;
    }else if(al[position][1]==al[position+1][1]){
      al[position-1][1]=al[position][1];
      al[position+1][1]=0;
    }
  }
  public static void shiftRight(int position){          //Bewegt das Auto im Array um 1 nach rechts
    if(al[position][1]==al[position+1][1]){
      al[position+2][1]=al[position][1];
      al[position][1]=0;
    }else if(al[position][1]==al[position-1][1]){
      al[position+1][1]=al[position][1];
      al[position-1][1]=0;
    }
  }
  public static void setAl(char[][] a){               //Überschreibt das Globale Array al mit a, damit
                                                      //man nicht
    for(int y=0; y<i;y++){                            //mit einem Bearbeiteten Array weiterarbeitet
      al[y][0]=a[y][0];
      al[y][1]=a[y][1];
    }
  }
```