Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)



Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_Программное обеспечение ЭВМ и информационные технологии\_\_\_\_\_\_

**Отчет по практике**

Студент \_\_\_\_\_\_\_\_\_ Турсунов Жасурбек Рустамович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Группа \_\_ИУ7-16\_\_

Название практики \_\_\_\_\_\_\_\_\_\_\_\_\_Сканер\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель

Преподаватель\_ \_Кузнецова О.В.\_ \_\_\_\_\_\_\_\_\_

Должность ФИО Подпись

Москва, 2018

Оглавление отчета:

1)Условие задачи.

2)Схема программы.

3)Описание программы.

4)Текст программы.

5)Заключение.

6)Список использованной литературы.

1)Условие задачи:

«Сканер»

Задание:

Сканер работает, просматривая последовательность горизонтальных слоёв тела. Из этих слоёв может быть сформирована исходная трёхмерная модель. Нужно написать программу, для изображения каждого слоя, использую данные, полученные во время сканирования.

Сканер состоит из четырёх массивов датчиков, установленных вокруг матрицы 5х7. Первый массив состоит из 5 датчиков, направленных вправо, второй массив имеет 11 датчиков, направленных по диагонали к правому верхнему углу, третий массив имеет 7 датчиков, направленных вверх, и четвертый массив 11 датчика, направленных по диагонали к левому верхнему углу. Каждый датчик делает запись толщины той части объекта, которая расположена непосредственно перед ним.

Результаты чтения и номера массивов датчиков зафиксированы против часовой стрелки.

Ввод данных:

Исходный файл начинается с номера слоя. Для каждого слоя, написаны строки с 5, 11, 7, 11 целыми числами. Порядок чтения обозначен в диаграмме. Хотя результат сканирования может быть и неоднозначным, но ваши данные будут иметь только один вариант.

Вывод результатов:

Для каждого слоя, программа должна напечатать 5 линий по 7 ячеек. Если ячейка представляет часть объекта, то печатать ''#'', в противном случае ''.''. Между слоями напечатать пустую строку

Пример ввода Пример вывода

3 3 5 6 2 . # # . # . .

0 1 2 2 3 32 3 2 1 0 . # # . # . .

0 4 4 3 5 2 1 . # # # # # .

0 0 1 3 3 4 4 3 1 0 0 . # # # # # #

. . # # . . .

3 3 5 6 2 . # # . # . .

0 1 2 2 3 3 2 3 2 1 0 . # # . # . .

0 4 4 3 5 2 1 . # # # # # .

0 0 1 3 3 4 4 3 1 0 0 . # # # # # #

. . . # # . .

2)Схема программы:

C:\Users\Jasur\Desktop\graph.png

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

C:\Users\Jasur\Desktop\defselect.png

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

C:\Users\Jasur\Desktop\defcheckmax.png

C:\Users\Jasur\Desktop\defcheck.png\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

C:\Users\Jasur\Desktop\defoutput.png

3) Описание алгоритма решения:

1. За основу решения задачи был выбран метод «Поиска с возвратом» или же «Backtracking».

Решение задачи методом поиска с возвратом сводится к последовательному расширению частичного решения. Если на очередном шаге такое расширение провести не удается,

то возвращаются к более короткому частичному решению и продолжают поиск дальше. Данный алгоритм позволяет найти все решения поставленной задачи, если они существуют.

1. Следуя указанию условия задачи, нужно было пройтись сначала по первому массиву, датчики которого направлены в вправо. Затем нужно было пройтись по второму массиву, датчики которого направлены к правому верхнему углу. Аналогично и четвёртым массивом, но только там датчики направлены к левому верхнему углу. А вот датчики третьего массива были направлены вертикально вверх. Каждая цифра массива показывает количество «#» в каждом раду.
2. Основной задачей функции def output(arr) является ставка символов «#» и «.» при выполнении условия. И это всё должно происходить циклически. Функция select(i, j) выбирает возможные случаи нулей и единиц, которые можно поставить в каждом ряду, если удовлетворяются все условия. В свою очередь функция def checkmax(I,j) возвращает True, когда в ячейке должна быть 1, иначе возвращает False , если в ячейке может быть 0.

4)Текст программы:

'''a = [ [10,10,6,4,6,8,13,15,11,6,],

[0,1,2,2,2,2,4,5,5,6,7,6,5,6,6,5,5,6,6,3,2,2,1,0,],

[2,4,5,5,7,6,7,10,10,10,7,3,3,5,5,], #Исходные данные(Ввод)

[0,0,0,0,2,4,4,6,9,9,8,8,7,5,4,3,4,4,4,4,3,1,0,0,],

]'''

a = [ [3,3,2,2,4,3,],

[1,0,1,1,3,1,2,1,2,3,0,1,1], #Для проверки

[2,2,2,2,2,2,2,3],

[0,0,2,2,2,3,1,2,2,1,0,2,0,]

]

'''a = [ [3,3,5,6,2,],

[0,1,2,2,3,3,2,3,2,1,0,], #Для проверки

[0,4,4,3,5,2,1,],

[0,0,1,3,3,4,4,3,1,0,0,]

]'''

def output(arr):

for i in arr:

for j in i:

if j == 0: print('.', end = ' ') #Если попадается часть объекта, то ставим "#", в противном случае "."

else: print('#', end = ' ')

print()

def checkmax(i, j):

boo = (c[0][i] == n) or (c[2][j] == m)

if boo:

return True

if (i + j) <= m - 1:

boo = (c[1][i + j] == i + j + 1)

elif (i + j) >= n - 1:

boo = (c[1][i + j] == m + n - (i + j + 1))

else :

boo = (c[1][i + j] == m)

if boo: return boo

if (i - j + n - 1) <= m - 1:

boo = c[3][i - j + n - 1] == i - j + n

elif (i - j + n - 1) >= n - 1:

boo = c[3][i - j + n - 1] == m + n - (i - j + n)

return boo

def check(i,j):

return (a[0][i] > 0) and (a[2][j] > 0) and (a[1][i + j] > 0) and (a[3][i - j + n - 1] > 0)

def select(i,j):

global count, a, result, count0, n, m, find #Глобальные переменные для использования их в программе

if j > n - 1:

i += 1

j = 0

if count == 0:

output(result)

find = True

return None

elif (i <= m - 1) and ((n \* (m - 1 - i) + (n - j)) >= count) and (not find) : #2 скобка считает количество ячеек после точки,

#1 скобка чтобы не переходила на следующую строчку

# not select cell (i,j)

if (not (checkmax(i, j))) and (count0 > 0):

count0 -= 1

result[i][j] = 0 #Счетчик count0 считает количество нулей

select(i, j + 1)

count0 += 1

if (check(i, j) and count > 0) and (not find) :

#select cell (i,j)

result[i][j] = 1

count -= 1

a[0][i] -= 1

a[2][j] -= 1

a[1][i + j] -= 1

a[3][i - j + n - 1] -= 1 #Счетчик count считает количество единиц

select(i, j + 1)

result[i][j] = 0

count += 1

a[0][i] += 1

a[2][j] += 1

a[1][i+j] += 1

a[3][i - j + n - 1] += 1

f = open('305.txt','r')

k = int(f.readline().strip())

for l in range(k):

a = [ ]

a.append(list(map(int, f.readline().split())))

a.append(list(map(int, f.readline().split())))

a.append(list(map(int, f.readline().split()))) #Работа с файлом

a.append(list(map(int, f.readline().split())))

c = [ i for i in a]

n = len(a[2])

m = len(a[0])

result = [[0 for i in range(n)] for j in range (m)]

count = sum (a[0])

count0 = n \* m - count

find = False

b = [[0 for i in range(m)],

[0 for i in range(m + n - 1)],

[0 for i in range(n)],

[0 for i in range(m + n - 1)],

]

select(0, 0)

f.close()

5) Заключение:

При выполнении этой работы был изучен новый для меня метод, так называемый метод «Backtracking». Метод поиска с возвратом является универсальным. Однако время нахождения решения может быть очень велико даже при небольших размерностях задачи (количестве исходных данных), причём настолько велико (может составлять недели или даже года), поэтому при проектировании таких алгоритмов, обязательно нужно теоретически оценивать время их работы на конкретных данных.

Практическое применение. Данная задача(Сканер) нужна для сканирования всех горизонтальных слоёв тела, и на основе этих данных может быть сформирована 3-D модель объекта.

6)Список использованной литературы:

1. Дмитриева М. В. «Решение задач методом перебора с возвратом», 2005;

2. Steven S. Skiena «Backtracking Chapter 8», New York, 2003;