

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту

Кафедра математичного моделювання та аналізу даних

ПРАКТИЧНА РОБОТА №3

на тему: «»

Виконав: студент 3 курсу групи НОМЕР
КС32
Спеціальності 122 «Комп'ютерні науки»
ПІБ Бабенко Андрій Андрійович

Харків – 2024

1. Написати програму яка приймає на вхід "пиріг з родзинками"

2. cake =

3.

4. ..o.....

5. ...o....

6.

7. // o это изюм

8. Нам потрібно рівно розрізати пиріг на n маленьких прямокутних шматочків так, щоб у кожному маленькому пирізі була 1 родзинка. n не є аргументом, ця кількість родзинок усередині торта

9. cake =

10.

11. ..o.....

12. ...o....

13.

14.

15. Результат ось такий масив

16. [

17.

18. ..o.....

19. ,

20. ...o....

21.

22.]

Кількість родзинок завжди більше 1 та менше 10. Якщо рішень кілька, виберіть те, що має найбільшу ширину першого елемента масиву. Поступово розрізати на n частин, що означає однакову площу. Але їхня форма може бути різною. Кожен шматок торта має бути прямокутним. Приклад

пирог =

.o.....

.....o.

....o...

..o.....

У цьому прикладі ми можемо знайти три рішення:
рішення 1 (горизонтальне різання):

[

.o..... //кусок 1

,

.....o. //кусок 2

,

....o... //кусок 3

,

..o..... //часть 4

]

рішення 2 (вертикальне різання):

[

.o //кусок 1

..

..

..

,

.. //шт. 2

..

..

o.

```

,
.. //шт. 3
..
о.
..
,
.. //шт. 4
о.
..
..
]

```

решение 3

```

[
.о.. //шт. 1
....
,
.... //шт. 2
..о.
,
.... //шт. 3
..о.
,
о... //шт. 4
....
]

```

нам потрібно вибрати рішення 1 як результат
Приклад різних форм:

торт =

```

.о.о....
.....
....о...
.....
.... о..
.....

```

результат должен быть:

```

[
.о // штук 1
..
..
..
..
..
,
.о.... //шт. 2
.....
,
..о... //шт. 3
.....
,
...о.. //шт. 4
.....
]

```

```
# Функція для знаходження координат родзинок у пирозі
def find_raisins(cake)

  raisins = []

  cake.each_with_index do |row, i|

    row.chars.each_with_index do |char, j|

      raisins << [i, j] if char == 'o'

    end

  end

  raisins

end

# Функція для перевірки, чи шматок пирога містить лише одну родзинку
def has_one_raisin?(piece)

  piece.flatten.count('o') == 1

end

# Функція для розділення пирога на прямокутні частини
def split_cake(cake, raisins)

  return [cake] if raisins.size == 1 # Якщо одна родзинка, не потрібно різати

  best_split = nil

  max_first_width = 0

  # Перебираємо всі можливі горизонтальні розрізи
```

```
(1...cake.size).each do |i|  
  top = cake[0...i]  
  bottom = cake[i..-1]  
  
  top_raisins = find_raisins(top)  
  bottom_raisins = find_raisins(bottom)  
  
  next if top_raisins.empty? || bottom_raisins.empty? # Пропускаємо, якщо  
частина без родзинок  
  
  top_split = split_cake(top, top_raisins)  
  bottom_split = split_cake(bottom, bottom_raisins)  
  
  if top_split && bottom_split  
    current_split = top_split + bottom_split  
    current_width = top_split[0][0].length # Ширина першого шматка  
  
    if current_width > max_first_width  
      max_first_width = current_width  
      best_split = current_split  
    end  
  end  
end  
  
# Перебираємо всі можливі вертикальні розрізи  
(1...cake[0].size).each do |j|
```

```
left = cake.map { |row| row[0...j] }
```

```
right = cake.map { |row| row[j..-1] }
```

```
left_raisins = find_raisins(left)
```

```
right_raisins = find_raisins(right)
```

```
next if left_raisins.empty? || right_raisins.empty? # Пропускаємо, якщо частина  
без родзинок
```

```
left_split = split_cake(left, left_raisins)
```

```
right_split = split_cake(right, right_raisins)
```

```
if left_split && right_split
```

```
  current_split = left_split + right_split
```

```
  current_width = left_split[0][0].length # Ширина першого шматка
```

```
  if current_width > max_first_width
```

```
    max_first_width = current_width
```

```
    best_split = current_split
```

```
  end
```

```
end
```

```
end
```

```
best_split
```

```
end
```

```
# Основна функція для запуску програми
```

```
def solve(cake)
```

```
  raisins = find_raisins(cake)
```

```
  result = split_cake(cake, raisins)
```

```
  puts "Результат поділу:"
```

```
  result.each_with_index do |piece, i|
```

```
    puts "Шматок #{i + 1}:"
```

```
    piece.each { |row| puts row }
```

```
    puts
```

```
  end
```

```
end
```

```
# Приклад використання
```

```
cake = [
```

```
  '.....',
```

```
  '..O.....',
```

```
  '...O.....',
```

```
  '.....'
```

```
]
```

```
solve(cake)
```

Код програми

```
main.rb
1 # Функція для знаходження координат родзинок у пирозі
2 def find_raisins(cake)
3   raisins = []
4   cake.each_with_index do |row, i|
5     row.chars.each_with_index do |char, j|
6       raisins << [i, j] if char == 'o'
7     end
8   end
9   raisins
10 end
11
12 # Функція для перевірки, чи шматок пирога містить лише одну родзинку
13 def has_one_raisin?(piece)
14   piece.flatten.count('o') == 1
15 end
16
17 # Функція для розділення пирога на прямокутні частини
18 def split_cake(cake, raisins)
19   return [cake] if raisins.size == 1 # Якщо одна родзинка, не потрібно різати
20
21   best_split = nil
22   max_first_width = 0
23
24   # Перебираємо всі можливі горизонтальні розрізи
25   (1...cake.size).each do |i|
26     top = cake[0...i]
27     bottom = cake[i..-1]
28
29     top_raisins = find_raisins(top)
30     bottom_raisins = find_raisins(bottom)
```

input

Результат поділу:

Шматок 1:

```
.....
..o.....
```

Шматок 2:

```
...o.....
.....
```

...Program finished with exit code 0
Press ENTER to exit console.

Малюнок 1 – результат програми