**Міністерство освіти і науки України**

**Національний технічний університет України**

**“Київський політехнічний інститут”**

**ІМ. СІКОРСЬКОГО**

ІПСА

Кафедра Системного проектування

Лабораторна робота №3

## «ПРИЙНЯТТЯ РІШЕНЬ

## В УМОВАХ ПОВНОЇ ІНФОРМАЦІЇ (ЗАДАЧА ПРО УПАКУВАННЯ В КОНТЕЙНЕРИ)»

Виконала:

студентка групи ДА-42

Балан Катерина

Київ – 2017

## Мета робот:

Ознайомитись з методами прийняття рішень в умовах повної інформації на прикладі задачі про упакування в контейнери та дослідити особливості їх використання.

## Варіант 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | С | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 1 | 1 | 83 | 86 | 77 | 15 | 93 | 35 | 86 | 92 | 49 | 21 | 62 | 27 | 90 | 59 | 63 | 26 | 40 | 26 | 72 | 36 |
| 2 | 11 | 38 | 67 | 29 | 82 | 30 | 62 | 23 | 67 | 35 | 29 | 02 | 22 | 58 | 59 | 67 | 93 | 56 | 11 | 42 |
| 3 | 29 | 73 | 21 | 19 | 84 | 37 | 98 | 24 | 15 | 70 | 13 | 26 | 91 | 80 | 56 | 73 | 62 | 70 | 96 | 81 |

**Хід роботи**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дані | Аналітичний розрахунок (кількість контейнерів) | | | | | | | |
| 1 рядок | 11 | | | | | | | |
| 2 рядок | 9 | | | | | | | |
| 3 рядок | 11 | | | | | | | |
| 1+2+3 рядок | 31 | | | | | | | |
| Дані | Кількість контейнерів | | | | Обчислювальна складність | | | |
| Без впорядкування | | | | Без впорядкування | | | |
| NFA | FFA | WFA | BFA | NFA | FFA | WFA | BFA |
| 1 рядок | 15 | 14 | 14 | 14 | 20 | 127 | 143 | 130 |
| 2 рядок | 12 | 11 | 11 | 11 | 20 | 85 | 101 | 91 |
| 3 рядок | 17 | 14 | 14 | 14 | 20 | 121 | 138 | 125 |
| 1+2+3 рядок | 42 | 36 | 37 | 36 | 60 | 820 | 937 | 900 |
| Дані | З впорядкуванням | | | | З впорядкуванням | | | |
| NFA | FFA | WFA | BFA | NFA | FFA | WFA | BFA |
| 1 рядок | 14 | 13 | 13 | 13 | 20 | 140 | 162 | 154 |
| 2 рядок | 12 | 10 | 10 | 10 | 20 | 104 | 134 | 134 |
| 3 рядок | 14 | 12 | 12 | 12 | 20 | 142 | 188 | 190 |
| 1+2+3 рядок | 41 | 33 | 34 | 33 | 60 | 1130 | 1301 | 1535 |

**Результати роботи програми**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA: [83, 86, 77, 15, 93, 35, 86, 92, 49, 21, 62, 27, 90, 59, 63, 26, 40, 26, 72, 36]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NFA: [[83], [86], [77, 15], [93], [35], [86], [92], [49, 21], [62, 27], [90], [59], [63, 26], [40, 26], [72], [36]]

Comparations: 20

Conteiners: 15

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FFA: [[83], [86], [77, 15], [93], [35, 49], [86], [92], [21, 62], [27, 59], [90], [63, 26], [40, 26], [72], [36]]

Comparations: 127

Conteiners: 14

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WFA: [[83], [86], [77, 15], [93], [35, 49], [86], [92], [21, 62], [27, 59], [90], [63, 26], [40, 26], [72], [36]]

Comparations: 143

Conteiners: 14

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BFA: [[83], [86], [77, 15], [93], [35, 49], [86], [92], [21, 62], [27, 59], [90], [63, 26], [40, 26], [72], [36]]

Comparations: 130

Conteiners: 14

---------------------

SORTED:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NFA: [[93], [92], [90], [86], [86], [83], [77], [72], [63], [62], [59], [49, 40], [36, 35, 27], [26, 26, 21, 15]]

Comparations: 20

Conteiners: 14

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FFA: [[93], [92], [90], [86], [86], [83], [77], [72, 27], [63, 36], [62, 35], [59, 26], [49, 40], [26, 21, 15]]

Comparations: 140

Conteiners: 13

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WFA: [[93], [92], [90], [86], [86], [83], [77], [72, 26], [63, 27], [62, 35], [59, 36], [49, 40], [26, 21, 15]]

Comparations: 162

Conteiners: 13

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BFA: [[93], [92], [90], [86], [86], [83], [77], [72, 27], [63, 36], [62, 35], [59, 26], [49, 40], [26, 21, 15]]

Comparations: 154

Conteiners: 13

Process finished *with* exit code 0

**Лістинг програми**

<https://github.com/katebalan/Decision_theory/blob/master/lab_tpr3/something.py>

# laboratory work #3 for Decision theory

# Next Fit Algorithm

*def* next\_fit(*boxes*, *container\_capacity*, *sort* = False):

*if sort*:

boxes = sorted(*boxes*, reverse=True)

containers = [[]]

containers\_count = 0

sum\_container = 0

comparison\_count = 0

*for* box *in boxes*:

sum\_container += int(box)

comparison\_count += 1

*if* sum\_container <= *container\_capacity*:

containers[containers\_count].append(box)

*else*:

sum\_container = int(box)

containers.append([])

containers\_count += 1

containers[containers\_count].append(box)

*print* "\*" \* 15

*print* "NFA: {}".format(containers)

*print* "Comparations: {}".format(comparison\_count)

*print* "Conteiners: {}".format(len(containers))

# First Fit Algorithm

*def* first\_fit(*boxes*, *container\_capacity*, *sort* = False):

*if sort*:

boxes = sorted(*boxes*, reverse=True)

containers = [[]]

containers\_count = 0

containers\_weight = [0]

comparison\_count = 0

*for* box *in boxes*:

comparison\_count += 1

*if* (containers\_weight[containers\_count] + box) <= *container\_capacity*:

containers[containers\_count].append(box)

containers\_weight[containers\_count] += box

*else*:

placed = False

*for* i *in* range(containers\_count):

comparison\_count += 1

*if* (containers\_weight[i] + box) <= *container\_capacity*:

containers[i].append(box)

containers\_weight[i] += box

placed = True

*break*

comparison\_count += 1

*if not* placed:

containers.append([])

containers\_count += 1

containers[containers\_count].append(box)

containers\_weight.append(box)

*print* "\*" \* 15

*print* "FFA: {}".format(containers)

*print* "Comparations: {}".format(comparison\_count)

*print* "Conteiners: {}".format(len(containers))

# Worst Fit Algorithm

*def* worst\_fit(*boxes*, *container\_capacity*, *sort* = False):

*if sort*:

boxes = sorted(*boxes*, reverse=True)

containers = [[]]

containers\_count = 0

containers\_weight = [0]

comparison\_count = 0

*for* box *in boxes*:

comparison\_count += 1

*if* (containers\_weight[containers\_count] + box) <= *container\_capacity*:

containers[containers\_count].append(box)

containers\_weight[containers\_count] += box

*else*:

min\_weight = min(containers\_weight)

comparison\_count += len(containers\_weight)

min\_index = containers\_weight.index(min\_weight)

comparison\_count += 1

*if* (min\_weight + box) <= *container\_capacity*:

containers[min\_index].append(box)

containers\_weight[min\_index] += box

*else*:

containers.append([])

containers\_count += 1

containers[containers\_count].append(box)

containers\_weight.append(box)

*print* "\*" \* 15

*print* "WFA: {}".format(containers)

*print* "Comparations: {}".format(comparison\_count)

*print* "Conteiners: {}".format(len(containers))

# Best Fit Algorithm

*def* best\_fit(*boxes*, *container\_capacity*, *sort* = False):

*if sort*:

boxes = sorted(*boxes*, reverse=True)

containers = [[]]

containers\_count = 0

containers\_weight = [0]

comparison\_count = 0

*for* box *in boxes*:

comparison\_count += 1

*if* (containers\_weight[containers\_count] + box) <= *container\_capacity*:

containers[containers\_count].append(box)

containers\_weight[containers\_count] += box

*else*:

best\_fit = []

*for* iter *in* range(containers\_count):

comparison\_count += 1

*if* containers\_weight[iter] + box <= *container\_capacity*:

best\_fit.append(containers\_weight[iter])

comparison\_count += 1

*if not* best\_fit:

containers.append([])

containers\_count += 1

containers[containers\_count].append(box)

containers\_weight.append(box)

*else*:

max\_weight = max(best\_fit)

comparison\_count += len(best\_fit)

max\_index = containers\_weight.index(max\_weight)

containers[max\_index].append(box)

containers\_weight[max\_index] += box

*print* "\*" \* 15

*print* "BFA: {}".format(containers)

*print* "Comparations: {}".format(comparison\_count)

*print* "Conteiners: {}".format(len(containers))