

Sterowanie Procesami Dyskretnymi

Jeziorek Mateusz

Maciej Szymczak

22.03.2018r.

1. Algorytm Johnsona i problem przepływowy

2. Algorytm RPQ i sortR

1. Algorytm Johnsona i problem przepływowy

- 2 Zadania

```
Podaj ile jest zadan:
2
9 8
2 1

0->9
9->11
11->CZAS MASZYN 1

CZAS MASZYN 2-> 18

W sumie permutacji: 2

Process returned 0 (0x0)   execution time : 0.924 s
Press any key to continue.
```

- 3 Zadania

```
Podaj ile jest zadan:
3
9 8
2 1
12 11

0->9
9->21
21->23
23->CZAS MASZYN 1

CZAS MASZYN 2-> 33

W sumie permutacji: 6

Process returned 0 (0x0)   execution time : 0.893 s
Press any key to continue.
```

- 4 Zadania

```
Podaj ile jest zadan:
4
9 8
2 1
12 11
15 14

0->9
9->24
24->36
36->38
38->CZAS MASZYN 1

CZAS MASZYN 2-> 50

W sumie permutacji: 24

Process returned 0 (0x0)   execution time : 1.034 s
Press any key to continue.
```

- 5 Zadań

```

Podaj ile jest zadan:
5
9 8
2 1
12 11
15 14
6 17

0->6
6->18
18->33
33->42
42->44
44->CZAS MASZYNY 1

CZAS MASZYNY 2-> 62

W sumie permutacji: 120

Process returned 0 (0x0)   execution time : 1.107 s
Press any key to continue.

```

- 6 Zadań

```

Podaj ile jest zadan:
6
9 8
2 1
12 11
15 14
6 17
21 24

0->6
6->18
18->39
39->54
54->63
63->65
65->CZAS MASZYNY 1

CZAS MASZYNY 2-> 92

W sumie permutacji: 720

Process returned 0 (0x0)   execution time : 1.111 s
Press any key to continue.

```

- 7 Zadań

```

7
9 8
2 1
12 11
15 14
6 17
21 24
18 29

0->6
6->18
18->36
36->57
57->72
72->81
81->83
83->CZAS MASZYNY 1

CZAS MASZYNY 2-> 118

W sumie permutacji: 5040

Process returned 0 (0x0)   execution time : 1.352 s
Press any key to continue.

```

- 8 Zadań

```

Podaj ile jest zadan:
8
9 8
2 1
12 11
15 14
6 17
21 24
18 29
10 13

0->6
6->16
16->31
31->52
52->70
70->82
82->91
91->93
93->CZAS MASZYN 1

CZAS MASZYN 2-> 131

W sumie permutacji: 40320

Process returned 0 (0x0)   execution time : 0.915 s
Press any key to continue.

```

- 9 Zadań

```

Podaj ile jest zadan:
9
9 8
2 1
12 11
15 14
6 17
21 24
18 29
10 13
5 30

0->5
5->14
14->26
26->44
44->65
65->80
80->90
90->96
96->98
98->CZAS MASZYN 1

CZAS MASZYN 2-> 153

W sumie permutacji: 362880

Process returned 0 (0x0)   execution time : 1.125 s
Press any key to continue.

```

- 10 Zadań

```

Podaj ile jest zadan:
10
9 8
2 1
12 11
15 14
6 17
21 24
18 29
10 13
5 30
51 56

0->5
5->14
14->26
26->44
44->95
95->116
116->131
131->141
141->147
147->149
149->CZAS MASZYN 1

CZAS MASZYN 2-> 231

W sumie permutacji: 3628800

Process returned 0 (0x0)   execution time : 1.645 s
Press any key to continue.

```

- 11 Zadań

```

Podaj ile jest zadan:
11
9 8
2 1
12 11
15 14
6 17
21 24
18 29
10 13
5 30
51 56
2 4

0->2
2->8
8->18
18->33
33->54
54->105
105->123
123->135
135->144
144->149
149->151
151->CZAS MASZYN 1

CZAS MASZYN 2-> 229

W sumie permutacji: 39916800

Process returned 0 (0x0)   execution time : 7.839 s
Press any key to continue.

```

- 12 Zadań

```

Podaj ile jest zadan:
12
9 8
2 1
12 11
15 14
6 17
21 24
18 29
10 13
5 30
51 56
2 4
19 99

0->2
2->8
8->18
18->33
33->52
52->103
103->124
124->142
142->154
154->163
163->168
168->170
170->CZAS MASZYN 1

CZAS MASZYN 2-> 326

W sumie permutacji: 4.79002e+008

Process returned 0 (0x0)   execution time : 90.840 s
Press any key to continue.

```

Wnioski:

W programie uwzględniłem ścieżkę postępowania przy algorytmie Johnsona, która ustala kolejność wykonywania zadań na maszynach. Wraz ze wzrostem ilości zadań przegląd zupełny wydłuża się, znośną granicą jest jeszcze 12 zadań, dla których trwa on ok 90s. Zaplanowanie takiej organizacji może przysłużyć się w wielu dziedzinach, m.in. przy rozładunkach i załadunkach, tworzenia siatek lekcyjnych itp. Ułożenie tej kolejki ma duże znaczenie w planowaniu czasów maszyny.

2. Algorytm RPQ i sortR

W zadaniu jednomaszynowym RPQ został zaimplementowany przegląd zupełny oraz sortowanie danych po wartości R.

Dla przeglądu zupełnego została użyta funkcja wykonująca permutacje podanego zbioru. Dla każdej z permutacji liczony jest czas CMax czyli funkcja podająca całkowity czas wykonania zleceń. Wszystkie permutacje wraz z czasami wykonania są wyświetlane na ekranie:

```
10:5:7
13:6:26
15:3:9

C_max 47 t:24
13:6:26
10:5:7
15:3:9

C_max 45 t:27
10:5:7
15:3:9
13:6:26

C_max 50 t:24
15:3:9
10:5:7
13:6:26

C_max 55 t:29
15:3:9
13:6:26
10:5:7

C_max 50 t:29
13:6:26
15:3:9
10:5:7

C_max 45 t:27
```

Dla sortowania po R algorytm najpierw sortuje dane po wartości R, następnie wyświetla wartość funkcji CMax dla posortowanych danych. Posortowane dane są zapisywane do pliku tekstowego w ułożonej kolejności:

```
in03.txt
C_max 47 t:24
Podaj nazwe pliku do zapisu permutacji: out03
out03.txt
```

Poprawność działania algorytmu SortR została sprawdzona zarówno dla mniejszej jak i większej ilości danych. Jednak przegląd zupełny dla większej ilości danych zajmuje sporo czasu ze względu na ilość permutacji (Dla N zdań $N!$ permutacji).

Jak widać na załączonym przykładzie przegląd zupełny jest algorytmem optymalnym ponieważ zawsze znajdzie najmniejszą wartość funkcji CMax (w tym przypadku 45). Jednakże jest to algorytm kosztowny czasowo/

SortR jest w porównaniu z przeglądem zupełnym bardzo szybkim algorytmem, nie znajduje on jednak najlepszego rozwiązania. Jednakże dla naszych danych funkcja CMax waha się od 45 do 55 a SortR dał wynik 47 co może być wynikiem zadowalającym.