

<b>NAZWA KURSU:</b>		<b>PROWADZĄCY:</b>	
STEROWANIE PROCESAMI DYSKRETNymi		Dr inż. Mariusz Makuchowski	
<b>TEMAT:</b>			
ALGORYTM DYNAMICZNEGO PROGRAMOWANIA ZWRACAJĄCY SUMĘ WITI			
<b>GRUPA:</b>	<b>OCENA:</b>	<b>PROPONOWANA OCENA:</b>	
SŁAWOMIR ŻABA,  MATEUSZ WOJDYŁA		4.5	

## 1. Opis działania algorytmu dynamicznego programowania:

Algorytm przez nas opisywany jest algorytmem szeregującym dla kryterium  $\sum_{i=1}^N W_i * T_i$ , gdzie  $N$  to liczba zadań nieprzerywalnych do uszeregowania. Każde zadanie posiada czas wykonania na maszynie  $p_i$ , wagę  $w_i$  oraz oczekiwany termin zakończenia  $d_i$ . Każdemu uszeregowanemu już zadaniu wyliczamy karę na podstawie spóźnienia  $T_i$ , gdzie  $T_i = \max(C_i - d_i, 0)$  ( $C_i$  - czas zakończenia itego zadania). Algorytm wykonuje  $(2^N - 1)$  kroków, w którym znajduje wartość optymalną uszeregowania podzbiorów zbioru zadań. Każdy z kroków zapisujemy w  $N$  bitach. Algorytm wykonuje kroki od pierwszego do  $2^N - 1$ , w każdym wyznaczając minimalną wartość funkcji celu dla podzbioru wszystkich zadań. Poniżej przedstawiony jest przykład pierwszych trzech kroków algorytmu dla  $N = 4$ .

0001 =  $\max(0, p_1 - d_1) * w_1$  - czas zakończenia zadania jest równy czasowi wykonywania tego zadania

0010 =  $\max(0, p_2 - d_2) * w_2$  - podobnie jak w poprzednim punkcie

0011 =  $\min(0001 \text{ z zadaniem drugim na końcu, } 0010 \text{ z zadaniem pierwszym na końcu})$ .

Czas zakończenia ostatniego zadania  $C'$  jest zawsze stały i równy sumie czasów wykonania na maszynie zadania pierwszego i drugiego ze względu na fakt, że w obojętnie jakiej kolejności ułożymy te zadania, to i tak skończą się zawsze w jednakowym czasie. Z tego wynika, że do wartości funkcji celu zbioru 0001 dodajemy  $w_2 * (C' - d_2)$ , a do wartości funkcji celu zbioru 0010 dodajemy  $w_1 * (C' - d_1)$ . Z tych dwóch sum wybieramy najmniejszą i zapisujemy ją, jako minimalną wartość funkcji celu dla zbioru 0011. Postępując analogicznie z algorytmem, ostatnią wartością funkcji celu do wyliczenia będzie dla zbioru zadań 1111, czyli:

1111 =  $\min(1110 + \text{pierwsze zadanie na końcu, } 1101 + \text{drugie zadanie na końcu, } 1011 + \text{trzecie zadanie na końcu, } 0111 + \text{czwarte zadanie na końcu})$ . Obliczona najmniejsza wartość funkcji celu dla zbioru 1111 jest najoptymalniejszym rozwiązaniem zadania.

## 2. Dane wyjściowe programu:

### • Data.10:

Suma Witi: 766,  
Permutacja: 6 9 2 5 1 3 4 7 8 10,  
czas wykonania: 0.000235s

### • Data.11:

Suma Witi: 799,  
Permutacja: 6 9 2 11 5 1 3 7 4 8 10,  
czas wykonania: 0.000493s

### • Data.12:

Suma Witi: 742,  
Permutacja: 6 9 2 11 5 1 3 12 7 4 8 10,  
czas wykonania: 0.001051s

- **Data.13:**  
Suma Witi: 688,  
Permutacja: 6 9 5 2 11 1 3 12 4 7 8 10 13,  
czas wykonania: 0.00221s
- **Data.14:**  
Suma Witi: 497,  
Permutacja: 6 9 5 1 2 3 11 12 4 7 8 10 13 14,  
czas wykonania: 0.004578s
- **Data.15:**  
Suma Witi: 440,  
Permutacja: 6 9 5 1 2 3 11 4 12 7 8 10 13 14 15,  
czas wykonania: 0.009865s
- **Data.16:**  
Suma Witi: 423,  
Permutacja: 6 9 5 1 2 3 11 4 7 12 8 10 13 14 15 16,  
czas wykonania: 0.020086s
- **Data.17:**  
Suma Witi: 417,  
Permutacja: 6 9 5 1 2 3 11 4 7 12 8 10 13 14 15 16 17,  
czas wykonania: 0.041643s
- **Data.18:**  
Suma Witi: 405,  
Permutacja: 6 9 5 1 2 3 11 18 4 7 12 8 10 13 14 15 16 17,  
czas wykonania: 0.085432s
- **Data.19:**  
Suma Witi: 393,  
Permutacja: 6 9 5 1 2 3 4 11 18 7 8 12 10 13 14 15 16 19 17,  
czas wykonania: 0.178127s
- **Data.20:**  
Suma Witi: 897,  
Permutacja: 6 20 9 5 1 2 3 11 18 4 7 12 8 10 13 14 19 15 16 17,  
czas wykonania: 0.365014s

**Łączny czas: 0.708734s**