

SPD - Lab nr 2 - Sprawozdanie

Marcin Wiśniewski (226524) Piotr Kida (226237)

6 kwietnia 2018

Spis treści

1	Cel	2
2	Wyniki	2
3	Porównanie algorytmów	6
4	Wnioski	7

1 Cel

Zadaniem na laboratoria było utworzenie implementacji algorytmu Schrage o dowolnej złożoności obliczeniowej, algorytm ten miał za zadanie podać kolejność zadań π^s oraz C_{max} .

2 Wyniki

Algorytm sprawdzono dla kilkunastu różnych zestawów danych wejściowych, w sprawozdaniu natomiast przywołane zostaną wyniki dla czterech różnych zestawów danych wejściowych.

Pierwszy zestaw danych wejściowych, dla ułatwienia w dalszej części pracy będziemy referować do niego poprzez nazwę SCHRAGE1, składa się z 6. zadań:

Numer zadania	Czasy wykonywania zadań
1	1 5 9
2	4 5 4
3	1 4 6
4	7 3 3
5	3 6 8
6	4 7 1
C_{max}	32

Czas egzekucji programu dla podanego zestawu danych wejściowych wynosił 0.047s.

Kolejność optymalna π^s zwrócona przez algorytm: 1, 5, 3, 2, 4, 6.

Drugi zestaw danych wejściowych, dla ułatwienia w dalszej części pracy będziemy referować do niego poprzez nazwę SCHRAGE2, składa się z 10. zadań:

Numer zadania	Czasy wykonywania zadań
1	219 5 276
2	84 13 103
3	336 35 146
4	271 62 264
5	120 33 303
6	299 14 328
7	106 46 91
8	181 93 97
9	263 13 168
10	79 60 235
C_{max}	687

Czas egzekucji programu dla podanego zestawu danych wejściowych wynosił 0.043s.

Kolejność optymalna π^s zwrócona przez algorytm: 10, 5, 2, 8, 1, 4, 6, 9, 3, 7.

Trzeci zestaw danych wejściowych, dla ułatwienia w dalszej części pracy będziemy referować do niego poprzez nazwę SCHRAGE3, składa się z 20. zadań:

Numer zadania	Czasy wykonywania zadań
1	592 82 321
2	547 18 687
3	284 11 219
4	568 46 507
5	189 76 604
6	465 52 577
7	234 53 732
8	391 49 718
9	205 10 64
10	157 29 176
11	505 40 637
12	211 21 326
13	518 57 645
14	625 15 53
15	500 51 66
16	114 1 506
17	454 91 167
18	174 75 319
19	340 56 480
20	184 61 69
C_{max}	1299

Czas egzekucji programu dla podanego zestawu danych wejściowych wynosił 0.041s.

Kolejność optymalna π^s zwrócona przez algorytm: 16, 10, 18, 7, 5, 19, 8, 6, 2, 13, 11, 4, 12, 1, 3, 17, 20, 15, 9, 14.

Czwarty zestaw danych wejściowych, dla ułatwienia w dalszej części pracy będziemy referować do niego poprzez nazwę SCHRAGE4, składa się z 20. zadań:

Numer zadania	Czasy wykonywania zadań
1	162 52 241
2	103 68 470
3	39 38 340
4	394 34 400
5	15 86 700
6	144 73 536
7	51 52 403
8	233 68 23
9	183 17 641
10	728 18 640
11	667 80 92
12	57 21 76
13	35 37 386
14	567 71 618
15	226 5 629
16	162 80 575
17	588 45 632
18	556 23 79
19	715 8 93
20	598 45 200
C_{max}	1399

Czas egzekucji programu dla podanego zestawu danych wejściowych wynosił 0.057s.

Kolejność optymalna π^s zwrócona przez algorytm: 5, 7, 6, 9, 15, 16, 2, 4, 13, 3, 1, 18, 14, 17, 20, 10, 19, 11, 12.

3 Porównanie algorytmów

W tej części sprawozdania porównane zostają wyniki działania algorytmu Schrage oraz algorytmu z zajęć poprzednich, dla następujących zestawów danych wejściowych.

Wyniki dla zestawu danych SCHRAGE1:

C_{max} dla algorytmu Schrage	C_{max} poprzedni
32	34

Wyniki dla zestawu danych SCHRAGE2:

C_{max} dla algorytmu Schrage	C_{max} poprzedni
687	746

Wyniki dla zestawu danych SCHRAGE3:

C_{max} dla algorytmu Schrage	C_{max} poprzedni
1299	1576

Wyniki dla zestawu danych SCHRAGE4:

C_{max} dla algorytmu Schrage	C_{max} poprzedni
1399	1594

Dla każdego z wykorzystywanych zestawów danych algorytm Schrage uzyskuje czas C_{max} krótszy, w porównaniu do algorytmu, którego implementacją zajmowaliśmy się na poprzednich laboratoriach. Algorytm ten wykonuje swoje zadanie w czasie wielokrotnie mniejszym niż swój poprzednik.

4 Wnioski

Z przedstawionych w sprawozdaniu wyników, mając do dyspozycji znany nam wachlarz zaimplementowanych już algorytmów, można śmiało wywnioskować, że w przypadku problemu wokół którego operujemy, oraz dla zestawów danych rzędu zbliżonego do danych testowych, rozwiązaniem optymalnym będzie implementacja algorytmu Schrage. Wniosek ten nasuwa się głównie z dwóch powodów:

1. uzyskiwanie lepszej wartości C_{max} ,
2. krótszy czas egzekucji programu.

Racja przywołana w punkcie drugim związana jest bezpośrednio z jakością utworzonej implementacji, dokładne porównanie czasów wykonywania różnych podejść do rozwiązania problemu byłoby możliwe po uwczesniejszym upewnieniu się, że oba kody są znormalizowane względem tej samej metryki. Elementem równie interesującym z perspektywy wykonywanego zadania jest uzyskany czas C_{max} , którego wartość, dzięki zastosowaniu algorytmu Schrage, udało się zredukować w każdym z wymienionych przypadków. Ze spostrzeżeń płynących z powyższej pracy nasuwa się teza, zgodnie z którą algorytm Schrage jest najbardziej efektywnym sposobem rozwiązywania problemu kolejkovania zadań, ze zbioru dotychczas poznanych metod.