

Projektowanie efektywnych algorytmów - Projekt 1

Problem komiwojażera - metoda programowania dynamicznego.

Termin zajęć: Czwartek 09:15-11:00
Prowadzący: dr inż. Dariusz Banasiak
Adam Filipowicz 221713

1 Wstęp

W projekcie implementowałem metodę programowania dynamicznego dla problemu komiwojażera. Metoda ma złożoność czasową $O(n^2 \cdot 2^n)$, gdzie n to ilość miast (wierzchołków). Grafu używałem do reprezentacji odległości między miastami. Do reprezentacji grafu użyłem macierzy, w której przechowywane są wartości poszczególnych krawędzi, np. wartość $M[i][j]$ to wartość krawędzi między wierzchołkami (i, j) . W przykładach które znalazłem na stronie ['comopt.ifi.uni-heidelberg.de/software/TSPLIB95/'](http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/) wartości $M[i][j]$ i $M[j][i]$ to te same krawędzie (problem symetryczny). Napisany program testowałem na danych z podanej strony dla 17, 21 oraz 24 miast. Zużycie pamięci dla grafu to $O(V^2)$, gdzie V to ilość wierzchołków. Dodanie, sprawdzenie istnienia i usunięcie krawędzi mają złożoność $O(1)$. Sprawdzenie stopnia wierzchołka ma złożoność $O(V)$.

Wynik działania algorytmu przechowuję w tablicy o długości V w postaci znalezionej permutacji.

1.1 Problem komiwojażera

Problem komiwojażera polega na znalezieniu najkrótszego cyklu w grafie pełnym, gdzie każdy wierzchołek odwiedzamy dokładnie raz.

Algorytm Helda-Karpa

Zaimplementowany algorytm to algorytm Helda-Karpa (Bellmana-Helda-Karpa) rekurencyjnie. Jako element początkowy wybieram zawsze wierzchołek 0. Do przechowania wszystkich danych używam mapy. Jako klucz podaję wektor intów, czyli podzbiór wszystkich wierzchołków oprócz początkowego i na jego końcu wierzchołek w którym kończy się ruch. Jako wartość przechowuję strukturę dwóch intów: wartość minimalna ścieżki dla tego podzbioru oraz wierzchołek z którego przybyto. Obie liczby są potrzebne przy wyświetleniu wierzchołków. Wartość minimalna jest oczywiście potrzebna gdy podzbiór 'wyszukuje' elementu w mapie.

Przypadek prosty rekurencji jest gdy podzbiór ma 2 elementy (jeden wierzchołek oraz wierzchołek z tego podzbioru na którym kończy się ruch, czyli ten sam). W tym przypadku jako wartość minimalną przypisujemy ścieżkę z wierzchołka startowego do wierzchołka w podzbiorze.

Przypadek złożony, gdy podzbiór ma s elementów. Wtedy wartość minimalna

dla tego podzbioru to minimum z sumy wartości podzbiorów $n-1$ elementowych bez aktualnego końcowego wierzchołka dla różnych wierzchołków końcowych oraz odległości z tych końcowych wierzchołków do wierzchołka końcowego podzbioru n elementowego.

Opis ten lepiej zilustrują poniższe wzory:

Jeśli $s = 1$, to $D(S, p) = d_{1,p}$,

Jeśli $s > 1$, to $D(S, p) = \min_{x \in (S-p)} (D(S-p, x) + d_{x,p})$,

gdzie:

- S - podzbiór s -elementowy
- p - końcowy wierzchołek
- D - optymalna (minimalna) długość ścieżki
- d - odległość między wierzchołkami
- x - pewien wierzchołek (we wzorze należy do zbioru S bez

Przykład algorytmu dla 4 wierzchołków

Założmy że problem jest symetryczny opisany poniższą macierzą:

Miasta	1	2	3	4
1	0	41	13	64
2	42	0	55	40
3	13	55	0	14
4	64	40	14	0

Na początku wyznaczamy wartości $D(S, p)$ dla jednoelementowych zbiorów S :

- $D(2, 2) = d_{1,2} = 41[1]$
- $D(3, 3) = d_{1,3} = 13[1]$
- $D(4, 4) = d_{1,4} = 64[1]$

Następnie wyznaczamy wartości $D(S, p)$ dla dwuelementowych zbiorów:

- $D(\{2, 3\}, 2) = D(3, 3) + d_{3,2} = 13 + 55 = 68[3]$

- $D(\{2, 3\}, 3) = D(2, 2) + d_{2,3} = 41 + 55 = 96[2]$
- $D(\{2, 4\}, 2) = D(4, 4) + d_{4,2} = 64 + 40 = 104[4]$
- $D(\{2, 4\}, 4) = D(2, 2) + d_{2,4} = 41 + 40 = 81[2]$
- $D(\{3, 4\}, 3) = D(4, 4) + d_{4,3} = 64 + 14 = 78[4]$
- $D(\{3, 4\}, 4) = D(3, 3) + d_{3,4} = 13 + 14 = 27[3]$

W kwadratowych nawiasach jest poprzedni element czyli w mojej implementacji programowej wierzchołek z którego przybyło do podzbioru (część struktury w mapie).

Podobnie postępujemy w przypadku zbiorów trójelementowych:

- $D(\{2, 3, 4\}, 2) = \min(D(\{3, 4\}, 3) + d_{3,2}, D(\{3, 4\}, 4) + d_{4,2}) = \min(78 + 55, 27 + 40) = \min(133, 67) = 67[4]$
- $D(\{2, 3, 4\}, 3) = \min(D(\{2, 4\}, 2) + d_{2,3}, D(\{2, 4\}, 4) + d_{4,3}) = \min(104 + 55, 81 + 14) = \min(159, 95) = 95[4]$
- $D(\{2, 3, 4\}, 4) = \min(D(\{2, 3\}, 2) + d_{2,4}, D(\{2, 3\}, 3) + d_{3,4}) = \min(68 + 40, 96 + 14) = \min(108, 110) = 108[2]$

W ostatnim kroku bierzemy minimum z sumy 3-elementowych podzbiorów i wartości ścieżki do wierzchołka początkowego, czyli:

$$D(\{1, 2, 3, 4\}, 1) = \min(D(\{2, 3, 4\}, 2) + d_{2,1}, D(\{2, 3, 4\}, 3) + d_{3,1}, D(\{2, 3, 4\}, 4) + d_{4,1}) = \min(67 + 41, 95 + 13, 108 + 64) = \min(108, 108, 172) = 108[2].$$

Teraz, wracając po liczbach w kwadratowych nawiasach możemy odczytać minimalną ścieżkę: 1-2-4-3-1.

Podobnie dzieje się w mojej, rekurencyjnej implementacji. Wywołuję funkcję z pełnych zbiorów, oprócz wierzchołka początkowego (n wierzchołków). Dla każdego szukam wszystkich podzbiorów n-1 elementowych. Jeśli podzbiór istnieje już w mapie, odczytuję jedynie wartość. Jeśli nie istnieje, wywołuję rekurencyjną funkcję dla tego podzbioru. Do mapy dodaję zbiory tuż przed return w funkcji rekurencyjnej zarówno dla zbiorów 2-elementowych (osobny przypadek), jak i innych.

Ostateczny wygląd wszystkich optymalnych długości ścieżek D:

- $D(2, 2) = 41[1]$
- $D(3, 3) = 13[1]$
- $D(4, 4) = 64[1]$
- $D(\{2, 3\}, 2) = 68[3]$
- $D(\{2, 3\}, 3) = 96[2]$
- $D(\{2, 4\}, 2) = 104[4]$
- $D(\{2, 4\}, 4) = 81[2]$
- $D(\{3, 4\}, 3) = 78[4]$
- $D(\{3, 4\}, 4) = 27[3]$
- $D(\{2, 3, 4\}, 2) = 67[4]$
- $D(\{2, 3, 4\}, 3) = 95[4]$
- $D(\{2, 3, 4\}, 4) = 108[2]$
- $D(\{1, 2, 3, 4\}, 1) = 108[2]$

2 Plan eksperymentu

- Struktury danych są alokowane dynamicznie.
- Program testowany był dla znanych wyników dla ilości miast równej 17, 21 oraz 24 dla grafu symetrycznego oraz dla ilości miast równej 17 dla grafu niesymetrycznego.
- Każdy pomiar losowy powtórzony jest 10 razy za każdym razem dla nowej populacji, a następnie liczona jest średnia. Ilości wierzchołków wynoszą : 4, 6, 8, 10, 12, 14, 16, 18, 20.
- Do pomiarów czasu używałem funkcji QueryPerformanceCounter. W pętli zapisywałem czasy dla kolejnych populacji i liczyłem średnią.

- Program napisany został w języku C++ w środowisku Microsoft Visual Studio 2017.
- Waga krawędzi, jest liczbą z przedziału $(0, RAND_MAX)$ wylosowaną za pomocą funkcji `rand()`.
- Graf jest pełny zatem generuje wszystkie możliwe krawędzie (oprócz tych łączących wierzchołek z samym sobą). Graf może być symetryczny lub asymetryczny.

3 Zestawienie wyników

3.1 Testy dla znanych wyników

- gr17 : 2085
- gr21 : 2707
- gr24 : 1272

Wyniki ze strony dla podanej w nazwach plików ilości wierzchołków dla grafu symetrycznego.

br17: 39

Wynik ze strony dla 17 wierzchołków grafu asymetrycznego.

Grafy symetryczne

17 miast

```
Podaj nazwe pliku:C:\Adam\t17.txt
Krawedzie grafu:
0 633 257 91 412 150 80 134 259 505 353 324 70 211 268 246 121
633 0 390 661 227 488 572 530 555 289 282 638 567 466 420 745 518
257 390 0 228 169 112 196 154 372 262 110 437 191 74 53 472 142
91 661 228 0 383 120 77 105 175 476 324 240 27 182 239 237 84
412 227 169 383 0 267 351 309 338 196 61 421 346 243 199 528 297
150 488 112 120 267 0 63 34 264 360 208 329 83 105 123 364 35
80 572 196 77 351 63 0 29 232 444 292 297 47 150 207 332 29
134 530 154 105 309 34 29 0 249 402 250 314 68 108 165 349 36
259 555 372 175 338 264 232 249 0 495 352 95 189 326 383 202 236
505 289 262 476 196 360 444 402 495 0 154 578 439 336 240 685 390
353 282 110 324 61 208 292 250 352 154 0 435 287 184 140 542 238
324 638 437 240 421 329 297 314 95 578 435 0 254 391 448 157 301
70 567 191 27 346 83 47 68 189 439 287 254 0 145 202 289 55
211 466 74 182 243 105 150 108 326 336 184 391 145 0 57 426 96
268 420 53 239 199 123 207 165 383 240 140 448 202 57 0 483 153
246 745 472 237 528 364 332 349 202 685 542 157 289 426 483 0 336
121 518 142 84 297 35 29 36 236 390 238 301 55 96 153 336 0
```

Krawedzie grafu odczytanego z pliku.

```
Minimum: 2089
Ścieżka: 0 ( 246 ) 15 ( 157 ) 11 ( 95 ) 8 ( 338 ) 4 ( 227 ) 1 ( 289 ) 9 ( 154 ) 10 ( 110 ) 2 ( 53 ) 14 ( 57 ) 13 ( 96 ) 16 ( 35 ) 5 ( 34 ) 7 ( 29 ) 6 ( 47 ) 12 ( 27 )
3 ( 91 ) 0.
```

Znaleziona najmniejsza trasa oraz ścieżka razem z pojedynczymi drogami w nawiasach.

21 miast

```
Podaj nazwe pliku:C:\Adam\t21.txt
Krawedzie grafu:
0 510 635 91 385 155 110 130 490 370 155 68 610 655 480 265 255 450 170 240 380
510 0 355 415 585 475 480 500 605 320 380 440 360 235 81 480 440 270 445 290 140
635 355 0 605 390 495 570 540 295 700 640 575 705 585 435 420 755 625 750 590 495
91 415 605 0 350 120 78 97 460 280 63 27 520 555 380 235 235 345 160 140 280
385 585 390 350 0 240 320 285 120 590 430 320 835 750 575 125 650 660 495 480 480
155 475 495 120 240 0 96 36 350 365 200 91 605 615 440 125 370 430 265 255 340
110 480 570 78 320 96 0 29 425 350 160 48 590 625 455 200 320 420 220 205 350
130 500 540 97 285 36 29 0 390 370 175 67 610 645 465 165 350 440 240 220 370
490 605 295 460 120 350 425 390 0 625 535 430 865 775 600 230 680 690 600 515 505
370 320 700 280 590 365 350 370 625 0 240 300 250 285 245 475 150 77 235 150 185
155 380 640 63 430 200 160 175 535 240 0 90 480 515 345 310 175 310 125 100 240
68 440 575 27 320 91 48 67 430 300 90 0 545 585 415 205 265 380 170 170 310
610 360 705 520 835 605 590 610 865 250 480 545 0 190 295 715 400 180 485 390 345
655 235 585 555 750 615 625 645 775 285 515 585 190 0 170 650 435 215 525 425 280
480 81 435 380 575 440 455 465 600 245 345 415 295 170 0 475 385 190 405 255 105
265 480 420 235 125 125 200 165 230 475 310 205 715 650 475 0 485 545 375 395 380
255 440 755 235 650 370 320 350 680 150 175 265 400 435 385 485 0 225 87 205 280
450 270 625 345 660 430 420 440 690 77 310 380 180 215 190 545 225 0 315 220 165
170 445 750 160 495 265 220 240 600 235 125 170 485 525 405 375 87 315 0 155 305
240 290 590 140 480 255 205 220 515 150 100 170 390 425 255 395 205 220 155 0 150
380 140 495 280 480 340 350 370 505 185 240 310 345 280 105 380 280 165 305 150 0
```

Krawędzie grafu odczytanego z pliku.

```
Minimum: 2707
Ścieżka: 0 ( 68 ) 11 ( 27 ) 3 ( 63 ) 10 ( 100 ) 19 ( 155 ) 18 ( 87 ) 16 ( 150 ) 9 ( 77 ) 17 ( 180 ) 12 ( 190 ) 13 ( 170 ) 14 ( 105 ) 20 ( 140 ) 1 ( 355 ) 2 ( 295 ) 8
( 120 ) 4 ( 125 ) 15 ( 125 ) 5 ( 36 ) 7 ( 29 ) 6 ( 110 ) 0.
```

Znaleziona najmniejsza trasa oraz ścieżka razem z pojedynczymi drogami w nawiasach.

24 miast

```
Podaj nazwe pliku:C:\Adam\t24.txt
Krawedzie grafu:
0 257 187 91 150 80 130 134 243 185 214 70 272 219 293 54 211 290 268 261 175 250 192 121
257 0 196 228 112 196 167 154 209 86 223 191 180 83 50 219 74 139 53 43 128 99 228 142
187 196 0 158 96 88 59 63 286 124 49 121 315 172 232 92 81 98 138 200 76 89 235 99
91 228 158 0 120 77 101 105 159 156 185 27 188 149 264 82 182 261 239 232 146 221 108 84
150 112 96 120 0 63 56 34 190 40 123 83 193 79 148 119 105 144 123 98 32 105 119 35
80 196 88 77 63 0 25 29 216 124 115 47 245 139 232 31 150 176 207 200 76 189 165 29
130 167 59 101 56 25 0 22 229 95 86 64 258 134 203 43 121 164 178 171 47 160 178 42
134 154 63 105 34 29 22 0 225 82 90 68 228 112 190 58 108 136 165 131 30 147 154 36
243 209 286 159 190 216 229 225 0 207 313 173 29 126 248 238 310 389 367 166 222 349 71 220
185 86 124 156 40 124 95 82 207 0 151 119 159 62 122 147 37 116 86 90 56 76 136 70
214 223 49 185 123 115 86 90 313 151 0 148 342 199 259 84 160 147 187 227 103 138 262 126
70 191 121 27 83 47 64 68 173 119 148 0 209 153 227 53 145 224 202 195 109 184 110 55
272 180 315 188 193 245 258 228 29 159 342 209 0 97 219 267 196 275 227 137 225 235 74 249
219 83 172 149 79 139 134 112 126 62 199 153 97 0 134 170 99 178 130 69 104 138 96 104
293 50 232 264 148 232 203 190 248 122 259 227 219 134 0 255 125 154 68 82 164 114 264 178
54 219 92 82 119 31 43 58 238 147 84 53 267 170 255 0 173 190 230 223 99 212 187 60
211 74 81 182 105 150 121 108 310 37 160 145 196 99 125 173 0 79 57 90 57 39 182 96
290 139 98 261 144 176 164 136 389 116 147 224 275 178 154 190 79 0 86 176 112 40 261 175
268 53 138 239 123 207 178 165 367 86 187 202 227 130 68 230 57 86 0 90 114 46 239 153
261 43 200 232 98 200 171 131 166 90 227 195 137 69 82 223 90 176 90 0 134 136 165 146
175 128 76 146 32 76 47 30 222 56 103 109 225 104 164 99 57 112 114 134 0 96 151 47
250 99 89 221 105 189 160 147 349 76 138 184 235 138 114 212 39 40 46 136 96 0 221 135
192 228 235 108 119 165 178 154 71 136 262 110 74 96 264 187 182 261 239 165 151 221 0 169
121 142 99 84 35 29 42 36 220 70 126 55 249 104 178 60 96 175 153 146 47 135 169 0
```

Krawędzie grafu odczytanego z pliku.

```
Minimum: 1272
Ścieżka: 0 ( 54 ) 15 ( 84 ) 10 ( 49 ) 2 ( 59 ) 6 ( 25 ) 5 ( 29 ) 23 ( 36 ) 7 ( 30 ) 20 ( 32 ) 4 ( 40 ) 9 ( 37 ) 16 ( 39 ) 21 ( 40 ) 17 ( 86 ) 18 ( 68 ) 14 ( 50 ) 1 ( 43 )
19 ( 69 ) 13 ( 97 ) 12 ( 29 ) 8 ( 71 ) 22 ( 108 ) 3 ( 27 ) 11 ( 70 ) 0.
```

Znaleziona najmniejsza trasa oraz ścieżka razem z pojedynczymi drogami w nawiasach.

Graf asymetryczny

17 miast

```
Podaj czy graf jest asymetryczny (a) czy symetryczny (s):a
Podaj nazwe pliku:C:\Adam\ta17.txt
Krawedzie grafu:
0 3 5 48 48 8 8 5 5 3 3 0 3 5 8 8 5
3 0 3 48 48 8 8 5 5 0 0 3 0 3 8 8 5
5 3 0 72 72 48 48 24 24 3 3 5 3 0 48 48 24
48 48 74 0 0 6 6 12 12 48 48 48 48 74 6 6 12
48 48 74 0 0 6 6 12 12 48 48 48 48 74 6 6 12
8 8 50 6 6 0 0 8 8 8 8 8 8 50 0 0 8
8 8 50 6 6 0 0 8 8 8 8 8 8 50 0 0 8
5 5 26 12 12 8 8 0 0 5 5 5 5 26 8 8 0
5 5 26 12 12 8 8 0 0 5 5 5 5 26 8 8 0
3 0 3 48 48 8 8 5 5 0 0 3 0 3 8 8 5
3 0 3 48 48 8 8 5 5 0 0 3 0 3 8 8 5
0 3 5 48 48 8 8 5 5 3 3 0 3 5 8 8 5
3 0 3 48 48 8 8 5 5 0 0 3 0 3 8 8 5
5 3 0 72 72 48 48 24 24 3 3 5 3 0 48 48 24
8 8 50 6 6 0 0 8 8 8 8 8 8 50 0 0 8
8 8 50 6 6 0 0 8 8 8 8 8 8 50 0 0 8
5 5 26 12 12 8 8 0 0 5 5 5 5 26 8 8 0
```

Krawedzie grafu odczytanego z pliku.

```
Minimum: 39
Sciezka: 0 ( 0 ) 11 ( 5 ) 16 ( 0 ) 8 ( 0 ) 7 ( 12 ) 4 ( 0 ) 3 ( 6 ) 15 ( 0 ) 14 ( 0 ) 6 ( 0 ) 5 ( 8 ) 12 ( 0 ) 10 ( 0 ) 9 ( 0 ) 1 ( 3 ) 13 ( 0 ) 2 ( 5 ) 0.
```

Średnie wyniki

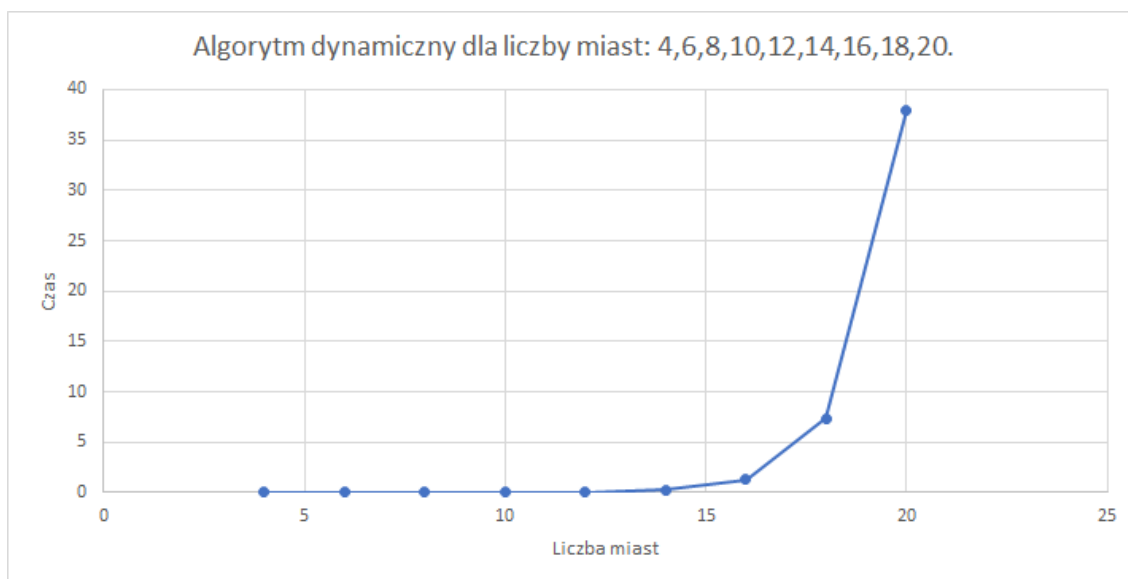
```
Algorytm dynamiczny symetryczny dla 17miast: 2.8666
Algorytm dynamiczny asymetryczny dla 17miast: 2.83433
Algorytm dynamiczny symetryczny dla 21miast: 86.3032
Algorytm dynamiczny symetryczny dla 24miast: 1018.76
```

Wszystkie podane czasy są w [s]. Powyższych wyników jest zbyt mało i liczba miast jest zbyt odległa od siebie i nierównomierna (17, 21, 24) do stworzenia wykresu, jednak z wyników możemy odczytać, że symetryczność grafu nie ma znaczenia dla czasu działania algorytmu.

3.2 Testy losowe

Wyniki czasu w tabeli są w [s].

Ilość miast	Czas
4	0,0001
6	0,0003
8	0,0034
10	0,0079
12	0,0361
14	0,2063
16	1,2350
18	7,3571
20	37,9662



Wykres został wygenerowany z wyników w programie Excel.

4 Wnioski

We wszystkich przypadkach złożoności uzyskane eksperymentalnie zgadzają się z teoretycznymi.

Wyniki wszystkich czterech zbadanych instancji (gr17, gr21, gr24 i br17) zgadzają się z podanymi na stronie internetowej. Zakładam że implementacja jest poprawna.

Z tabeli wyników widać że wyniki zgadzają się z teoretyczną złożonością (są mniej więcej 5-6 razy większe). Nieścisłości mogą wynikać z małej ilości próbek (10), losowości wyników lub niedokładności pomiaru czasu.

Z testów dla znanych wyników widać, że działanie dla grafu symetrycznego i asymetrycznego nie zmienia czasu działania algorytmu.

5 Bibliografia

1. Charles E. Leiserson. Ronald L. Rivest. Clifford Stein. Introduction to Algorithms. Third Edition. The MIT Press. Cambridge, Massachusetts London, England
2. A. Janiak, Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów, PLJ 1999.