実践的アルゴリズム理論
Theory of Advanced Algorithms

9. 動的計画法(3)

担当:上原隆平

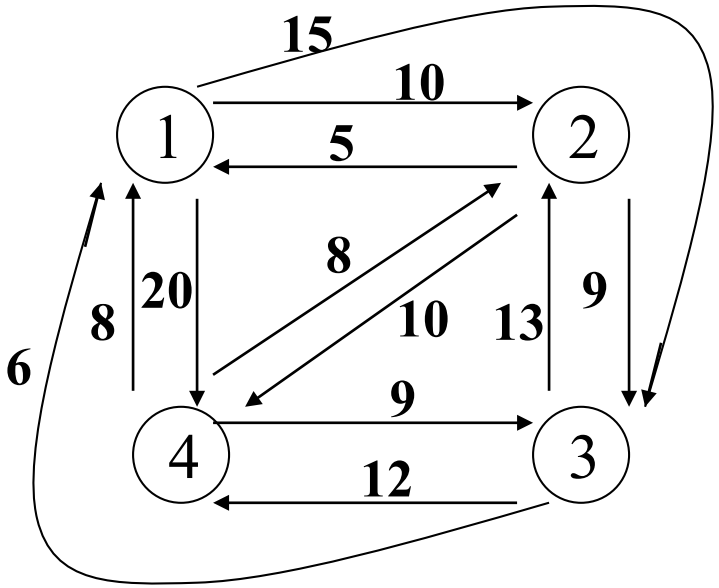
Theory of Advanced Algorithms
実践的アルゴリズム理論

9. Dynamic Programming (3)

Ryuhei Uehara

問題P28:（行商人問題）

n都市の相互を結ぶ道路網を表す重みつきグラフが与えられたとき，すべての都市を訪れて元の都市に戻ってくる周遊路の中で最短のものを求めよ.



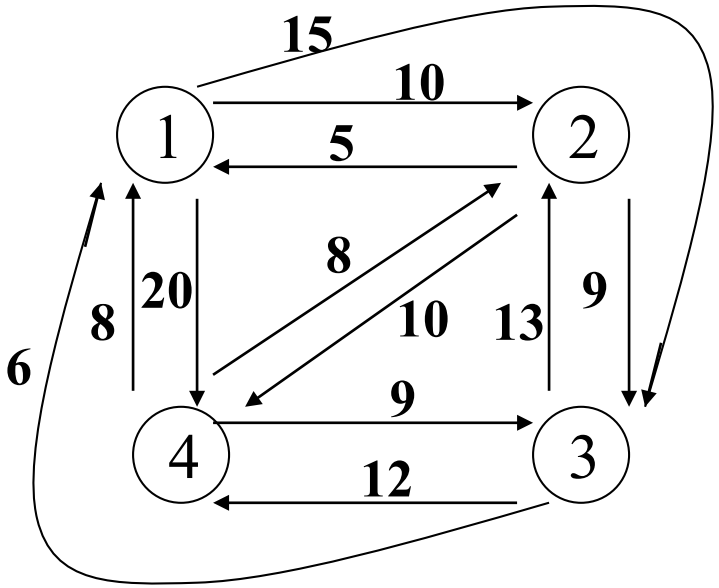
$$L=\begin{pmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{pmatrix}$$

都市iとjの間の距離を
L[i,j]とする.

周遊路(1,2,3,4,1) : 長さ=10+9+12+8=39,
周遊路(1,2,4,3,1) : 長さ=10+10+9+6=35,
周遊路(1,3,2,4,1) : 長さ=15+13+10+20=58,
周遊路(1,3,4,2,1) : 長さ=15+12+8+5=40, 等々

Problem P28: (Travelling-Salesperson Problem)

Given a weighted graph for a road network interconnecting n cities, find a shortest closed tour starting from a city and coming back to it after visiting every city.



$$L=\begin{pmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{pmatrix}$$

L[i,j] is the distance between city i and city j.

tour(1,2,3,4,1) : length=10+9+12+8=39,
tour(1,2,4,3,1) : length=10+10+9+6=35,
tour(1,3,2,4,1) : length =15+13+10+20=58,
tour(1,3,4,2,1) : length =15+12+8+5=40, etc.

周遊路は全部で何通りあるだろう.

どの2都市間にも道があるなら, (n-1)!通りの周遊路が存在.
Stirlingの公式によると

$$n! \fallingdotseq \sqrt{(2\pi n)}(n/e)^n$$

これは大雑把にはn^nという指数関数.

都市を1, 2, ..., nと番号づけ, 必ず都市1から出発して都市1に戻ってくるものとする.
全都市の集合を$N=\{1, 2, \dots, n\}$とし, その部分集合をSとする.
都市1を含まない部分集合Sに対して,

g(i, S) = 都市iから出発して, Sの都市をすべて通って都市1に戻る経路の中での最短路の長さ,
ただし, iはSに属さないこと,

と定めると, 求める最短周遊路の長さは

$g(1, \{2, 3, \dots, n\})$

として与えられることになる.

How many tours are there in total?

If there is a road between any two cities, then there are (n-1)! tours.
Using the Stirling's formula, we have

$$n! \fallingdotseq \sqrt{(2\pi n)}(n/e)^n .$$

This is roughly an exponential function n^n.

Numbering the cities as 1, 2, ... , n, and assume that we start at city 1 and come back to it.
The set of all cities: N={1, 2, ... , n}. S is a subset of N.
For a subset S not containing city 1,

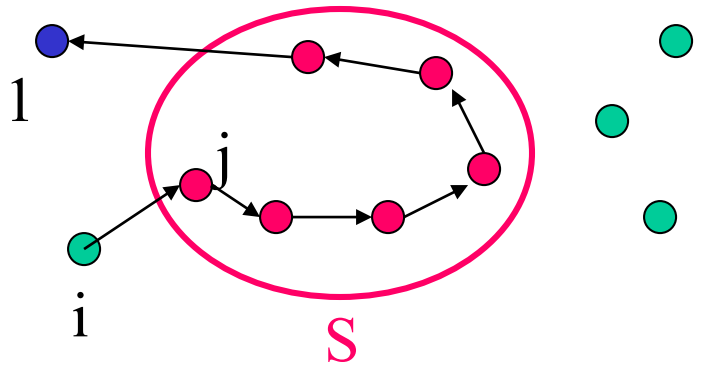
g(i, S) = the length of a shortest path from city i coming back to city 1 through every city of S.

Note that i does not belong to S.

Then, the length of the shortest tour is given as

g(1, {2, 3, ... , n}).

g(i, S)を再帰的に定義しよう



動的計画法を適用するためには，**部分問題に対する解が最適解に含まれるように最適解を再帰的に定義しなければならない.**

Sの中で都市iの次となる都市の候補をjとしたとき，
都市1までの最適な経路の長さは

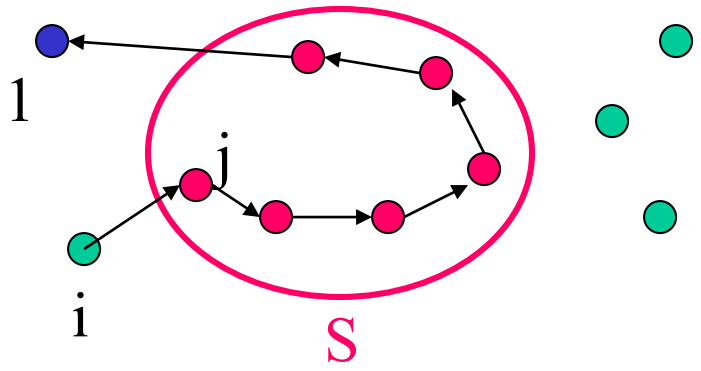
$$g(j, S-\{j\})$$

で与えられる．よって，g(i, S)に対する漸化式として

$$g(i, S) = \min_{j \in S}\{L[i,j] + g(j, S-\{j\})\}$$

を得る．ただし，iはSの要素ではない．
L[i,j]は都市i,j間の距離.

Let's define g(i, S) recursively!



To apply Dynamic Programming, an optimal solution must be defined recursively so that it includes a solution to a subproblem.

If j is a candidate among S for the city after city i, the length of an optimal path to city 1 is given by

$g(j, S-\{j\})$.

Thus, the recurrence equation for g(i, S) becomes

$$g(i, S) = \min_{j \in S}\{L[i,j] + g(j, S-\{j\})\},$$

where i is not an element of S.

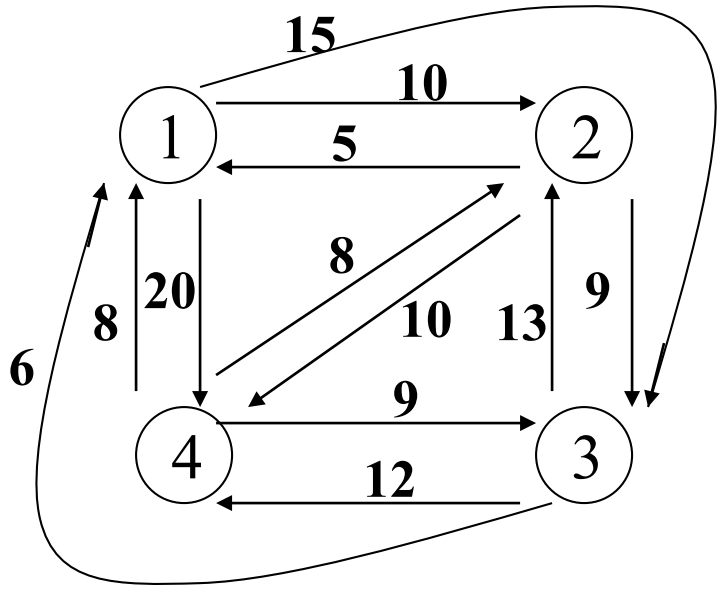
L[i,j] denotes the distance between city i and city j.

|S|=0

g(2, Φ)=L[2,1]=5
g(3, Φ)=L[3,1]=6
g(4, Φ)=L[4,1]=8

|S|=1

g(2, {3})=L[2,3]+g(3, Φ)=15
g(2, {4})=L[2,4]+g(4, Φ)=18
g(3, {2})=L[3,2]+g(2, Φ)=18
g(3, {4})=L[3,4]+g(4, Φ)=20
g(4, {2})=L[4,2]+g(4, Φ)=13
g(4, {3})=L[4,3]+g(3, Φ)=15



|S|=2

g(2, {3,4})=min{L[2,3]+g(3, {4}), L[2,4]+g(4,{3})}=min{29,25}=25
g(3, {2,4})=min{L[3,2]+g(2, {4}), L[3,4]+g(4,{2})}=min{31,25}=25
g(4, {2,3})=min{L[4,2]+g(2, {3}), L[4,3]+g(3,{2})}=min{23,27}=23

|S|=3

g(1, {2,3,4})=min{L[1,2]+g(2, {3,4}), L[1,3]+g(3, {2,4}),L[1,4]+g(4, {2,3})}
= min{35, 40, 43} = 35.

よって，最短周遊路の長さは35である.

|S|=0

g(2, Φ)=L[2,1]=5

g(3, Φ)=L[3,1]=6

g(4, Φ)=L[4,1]=8

|S|=1

g(2, {3})=L[2,3]+g(3, Φ)=15

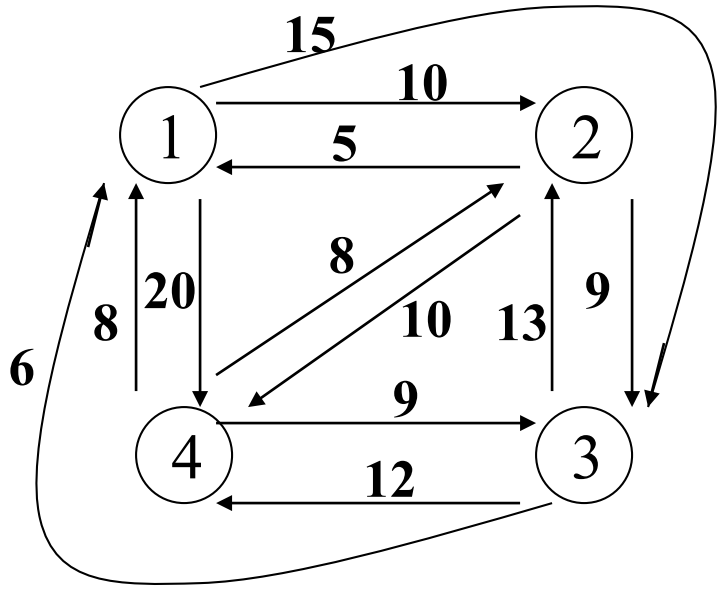
g(2, {4})=L[2,4]+g(4, Φ)=18

g(3, {2})=L[3,2]+g(2, Φ)=18

g(3, {4})=L[3,4]+g(4, Φ)=20

g(4, {2})=L[4,2]+g(4, Φ)=13

g(4, {3})=L[4,3]+g(3, Φ)=15



|S|=2

g(2, {3,4})=min{L[2,3]+g(3, {4}), L[2,4]+g(4,{3})}=min{29,25}=25

g(3, {2,4})=min{L[3,2]+g(2, {4}), L[3,4]+g(4,{2})}=min{31,25}=25

g(4, {2,3})=min{L[4,2]+g(2, {3}), L[4,3]+g(3,{2})}=min{23,27}=23

|S|=3

g(1, {2,3,4})=min{L[1,2]+g(2, {3,4}), L[1,3]+g(3, {2,4}),L[1,4]+g(4, {2,3})}

= min{35, 40, 43} = 35.

Therefore, the length of an optimal tour is 35.

アルゴリズムP28-A0:
入力：n都市間の距離を表すグラフ
U={1, 2, ... , n}とする.
for(i=2; i<=n; i++)
  g(i, Φ)=L[1,i];
for(k=1; k<n; k++){
  for 1を含まないサイズkのすべての部分集合S {
    for Sに含まれないすべてのiについて
      $g(i, S) = \min_{j \in S}\{L[i,j] + g(j, S-\{j\})\}$
}
return g(1, {2, 3, ... , n});

演習問題E10-4：上記のアルゴリズムの計算時間と記憶量をnの関数で表せ.（ヒント：多項式ではないがn^nよりは良い）

Algorithm P28-A0:

Input: A graph representing distances among cities.

U={1, 2, ... , n}.

```
for(i=2; i<=n; i++)
   g(i, Φ)=L[1,i];
for(k=1; k<n; k++){
   for each subset S of size k not containing 1 {
      for each i not contained in S
         g(i, S) = min_{j∈S}{L[i,j] + g(j, S-{j})}
}
return g(1, {2, 3, ... , n});
```

Exercise E10-4 : Express the computation time and amount of storage of the above algorithm as functions of n.
(Hint: It is not a polynomial, but it's better than n^n.)