

# 電気電子物理工学実験Ⅲ マイクロプロセッサ

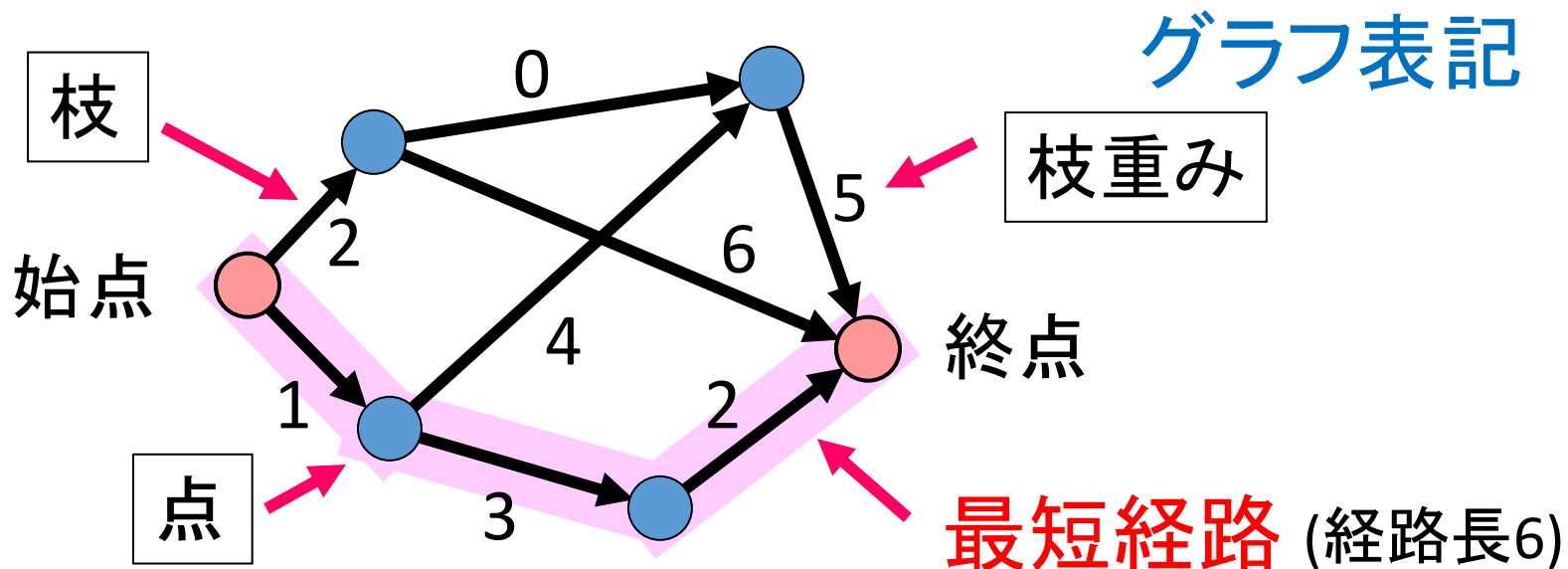
課題: 最短経路問題

埼玉大学工学部電気電子物理工学科

# ■最短経路問題

始点から終点へ行く最も短い経路

=枝重み合計が最小の経路を見つける



枝( $i, j$ )  $i$ : 枝の始点  
 $j$ : 枝の終点

$W_{ij}$ : 枝( $i, j$ )の重み

※枝には向きがあり、枝始点から枝終点にのみ進むことができる

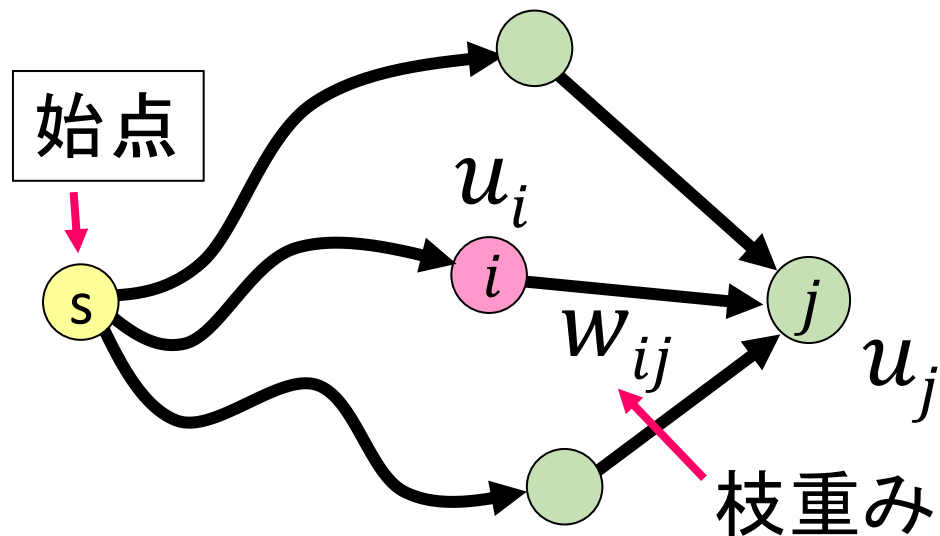
## ■ Bellman方程式

点 $j$ の最短経路長 $u_j$ について成り立つ関係式

$$u_s = 0$$

$$u_j = \min_{\forall (i,j)} \{u_i + w_{ij}\} \quad \forall j \neq s$$

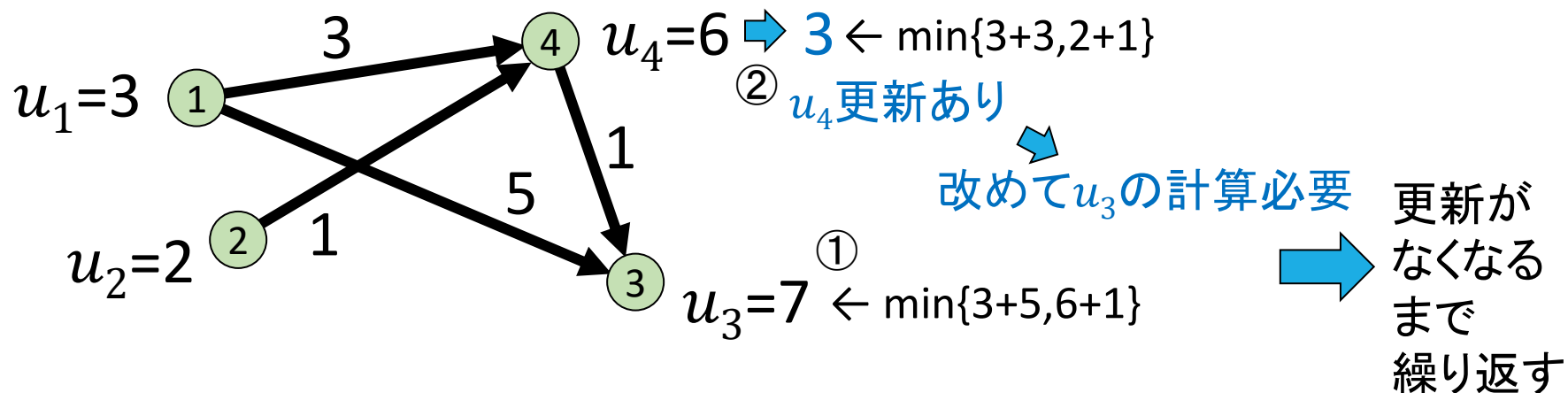
$j$ を終点とするすべての枝 $(i,j)$ について  
最小値(minimum)を求める



- Bellman方程式(連立方程式)を満たす $u_j$ を求めれば、それが最短経路長

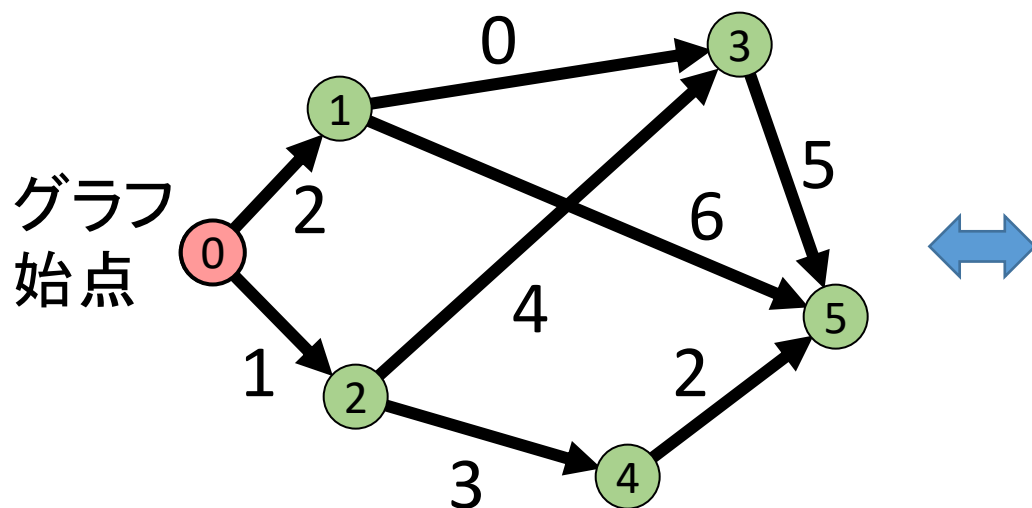
# ■ Bellman方程式を解く Bellman-Ford法

1. 点 $j$ の最短経路長 $u_j$ に適切な初期値を設定
  - ・ 本来は $+\infty$ (実際には十分大きな正の値で代用)
2. 全ての枝 $(i, j)$ について  
 $u_i + w_{ij}$ が $u_j$ よりも小さければ、その値で $u_j$ を更新  
 ( $u_j > u_i + w_{ij}$ ならば $u_j$ に $u_i + w_{ij}$ を代入)
3. 全点 $j$ について $u_j$ の更新がなければ終了  
 いずれかの点で更新があれば、2.に戻る



※終点だけでなく、すべての点の最短経路長が求まる

## ■ グラフのデータ



- ・ 枝の始点番号、終点番号、重みはそれぞれ1バイト
- ・ 始点番号、終点番号、重みの順に記録されているものとする
- ・ 点番号は0,1,2,...
- ・ 枝重みは0または正
- ・ 枝数は不定とし、始点番号が-1のとき、枝データの終了を表す

EDGES:	
db 0	# 枝(0,1)始点
db 1	# 枝(0,1)終点
db 2	# 枝(0,1)重み
db 0	# 枝(0,2)始点
db 2	# 枝(0,2)終点
db 1	# 枝(0,2)重み
db 1	# 枝(1,3)始点
db 3	# 枝(1,3)終点
db 0	# 枝(1,3)重み
db 1	# 枝(1,5)始点
db 5	# 枝(1,5)終点
db 6	# 枝(1,5)重み
:	
db 4	# 枝(4,5)始点
db 5	# 枝(4,5)終点
db 2	# 枝(4,5)重み
db -1	# 枝データ終了印

# ■ Bellman-Ford法のプログラム

L0:

UPDATE ← 0  
最初の枝を調べる準備

L1:

$i \leftarrow$  枝( $i, j$ )の始点

if(  $i < 0$  ) Jump to L2

$j \leftarrow$  枝( $i, j$ )の終点

$W \leftarrow$  枝( $i, j$ )の重み

if(  $u_j > u_i + W$  )then

$u_j \leftarrow u_i + W$

UPDATE ← 1

endif

Jump to L1

L2: if( UPDATE=1 ) Jump to L0

HLT

← 全ての枝を調べ終わった

← 経路長が更新(より小さくなる)されたら  
UPDATEを1にする

← UPDATE=1ならば、再度全枝を調べる

← UPDATE=0のままならば終了

UPDATE: db 0

I: db 0

J: db 0

W: db 0

U: db 0

db 99

db 99

:

db 99

←  $u_0$

←  $u_1$

←  $u_2$

← グラフの始点(0)の経路長は0

← その他の点の経路長の初期値は $\infty$   
十分大きな値として**99**を使用する

## ■ グラフの枝データの読み出し

L0:	:		●最初の枝を調べる準備
	LDA	X, EDGES ←	枝データの先頭アドレスをXレジスタに
	LD	EINDEX, X ←	そのアドレスをメモリ領域EINDEXに保存
	:		
L1:	LD	X, EINDEX	●1本の枝データを読み出し
	LD	AC, (X) ←	最初のデータ(枝始点番号)読み出し
	CPR	AC, 0 ←	枝始点番号を0と比較してSフラグを設定
	JS	L2 ←	負ならばジャンプ(全枝を調べ終わった)
	LD	I, AC ←	枝始点番号をメモリ領域Iに保存
	INC	X	
	LD	AC, (X)	
	LD	J, AC ←	枝終点番号をメモリ領域Jに保存
	INC	X	
	LD	AC, (X)	
	LD	W, AC ←	枝重みをメモリ領域Wに保存
	INC	X	
	LD	EINDEX, X ←	次の枝データ読み出しに備えてXを保存
	:		
EINDEX:	dw	0	
I:	db	0	
J:	db	0	
W:	db	0	

命令「LD AC, (X)」の追加定義必要  
(後で命令「LD (X), AC」も)

## ■ 点*i*の経路長 $u_i$ 参照

経路長データが配列Uに記録されているとして、*i*番目の要素U[*i*]を参照する

LDA	X, U	←	経路長配列Uの先頭アドレスをXレジスタに
LD	AC, I	←	点番号 <i>i</i> をACに読み出し
LD	WA, AC	←	Xと加算したいので準備としてWAに転送
ADD	WA, X	←	経路長配列Uの先頭アドレスと点番号 <i>i</i> を加算
LD	X, WA	←	加算結果(配列要素U[ <i>i</i> ]のアドレス)をXレジスタに転送
LD	AC, (X)	←	U[ <i>i</i> ]を読み出してACに保存
	:		
I:	db 0		
J:	db 0		
W:	db 0		
U:	db 0	←	U[0] = $u_0$
	db 99	←	U[1] = $u_1$
	db 99	←	U[2] = $u_2$
	:		
	db 99		



## ■ 点 $j$ の経路長 $u_j$ の更新

経路長データが配列Uに記録されているとして、J番目の要素U[J]を書き換える

LDA	X, U	←	経路長配列Uの先頭アドレスをXレジスタに
LD	AC, J	←	点番号JをACに読み出し
LD	WA, AC	←	Xと加算したいのでWAに転送
ADD	WA, X	←	経路長配列Uの先頭アドレスと点番号Iを加算
LD	X, WA	←	加算結果(配列要素U[J]のアドレス)をXレジスタに転送
LD	AC, U	←	$u_j$ を読み出してACに保存
LD	(X), AC	←	U[J]に $u_j$ を書き込み
:			
UJ:	db 0	←	$u_j$ の新しい値
I:	db 0		
J:	db 0		
W:	db 0		
U:	db 0	←	U[0] = $u_0$
	db 99	←	U[1] = $u_1$
	db 99	←	U[2] = $u_2$
:			
	db 99		

## ■追加命令定義

LD (X), AC

nbyte 1

opcode xxxxxxxx

0: MAR ← PC, PC ← inc

1: MDR ← mem

2: IR ← MDR

3: MAR ← X, MDR ← AC

4: WR = 1

Xレジスタの値をメモリアドレスとして  
メモリにACレジスタの値を書き込む

$MEM[X] \leftarrow AC$

← メモリアドレス: Xレジスタの値

← メモリに書き込むデータ: ACレジスタの値

← メモリに書き込み動作を行わせる

LD AC, (X)

nbyte 1

opcode xxxxxxxx

0: MAR ← PC, PC ← inc

1: MDR ← mem

2: IR ← MDR

:

Xレジスタの値をメモリアドレスとして  
メモリから読み出した値をACレジスタに保存

$AC \leftarrow MEM[X]$

→ ADD AC, (X)命令を参考に

$AC \leftarrow AC + MEM[X]$

(要は加算を行わなければよい)