

カレー作りの工程



作業番号

所要時間

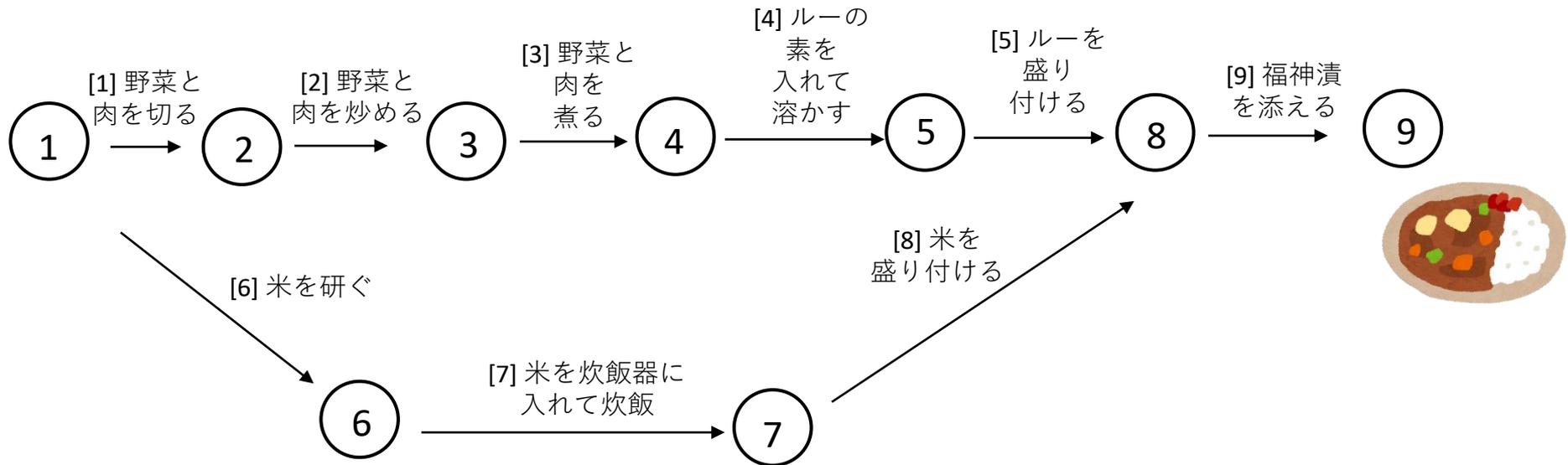
先行作業

[1] 野菜と肉を切る	5分	なし
[2] 野菜と肉を炒める	10分	[1]
[3] 野菜と肉を煮る	15分	[2]
[4] ルーの素を入れて溶かす	5分	[3]
[5] ルーを盛り付ける	1分	[4]
[6] 米を研ぐ	3分	なし
[7] 米を炊飯器に入れて炊飯	45分	[6]
[8] 米を盛り付ける	1分	[7]
[9] 福神漬を添える	1分	[5], [8]

合計 91分 (?)

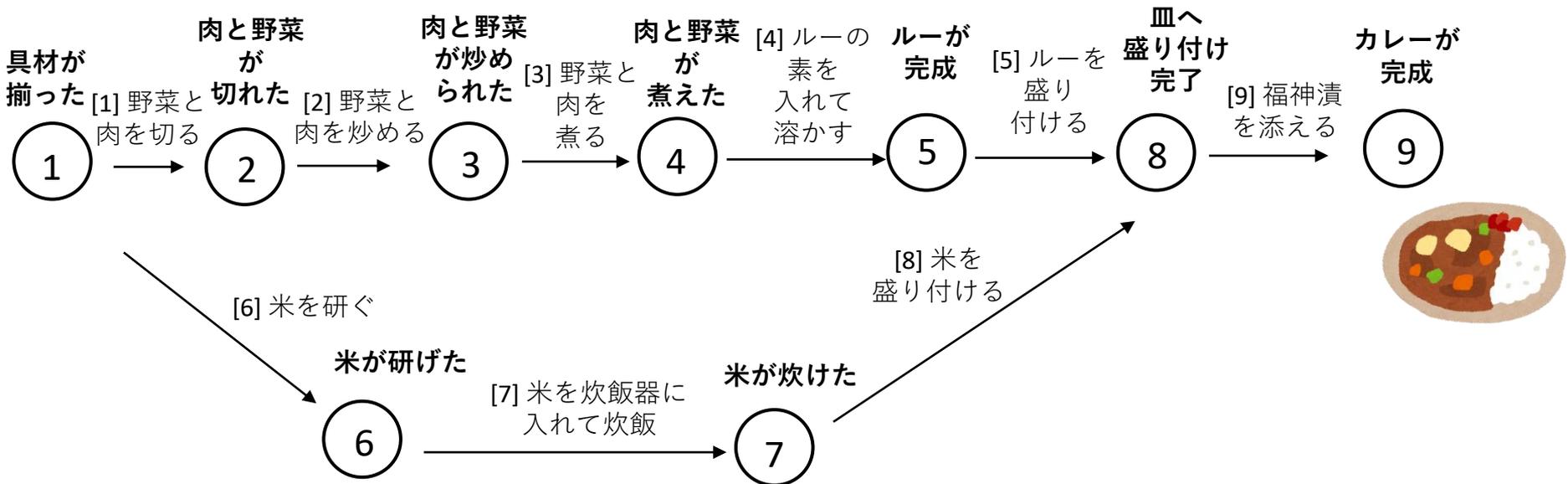
カレー作りのPERT図

(Program Evaluation and Review Technique)



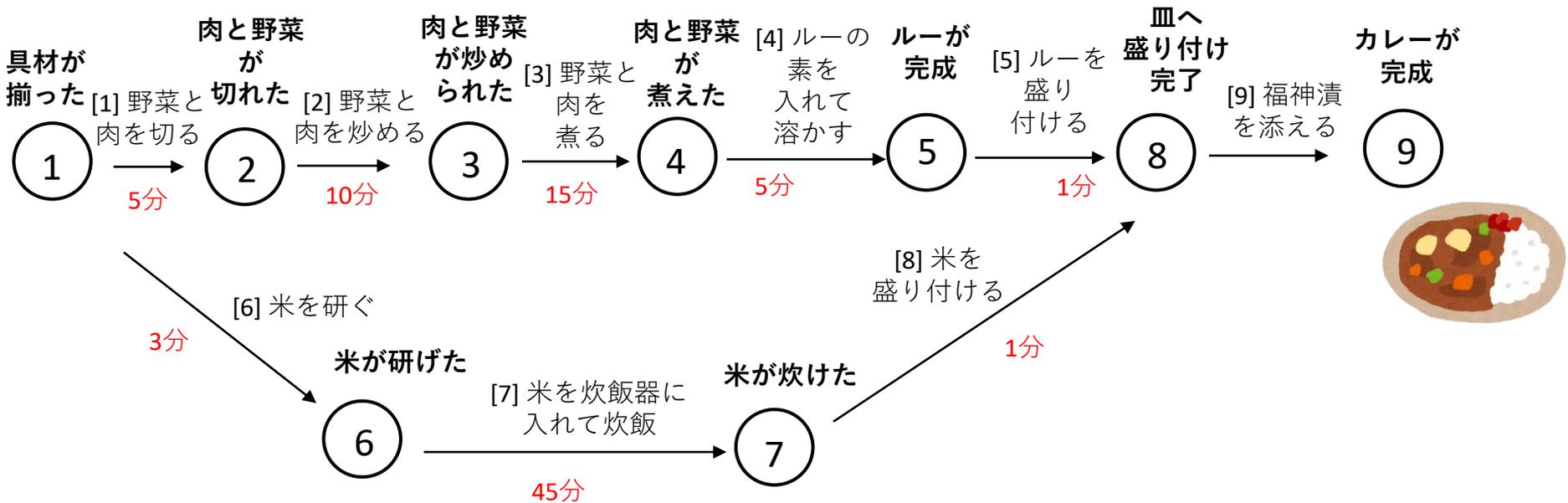
カレー作りのPERT図

(Program Evaluation and Review Technique)



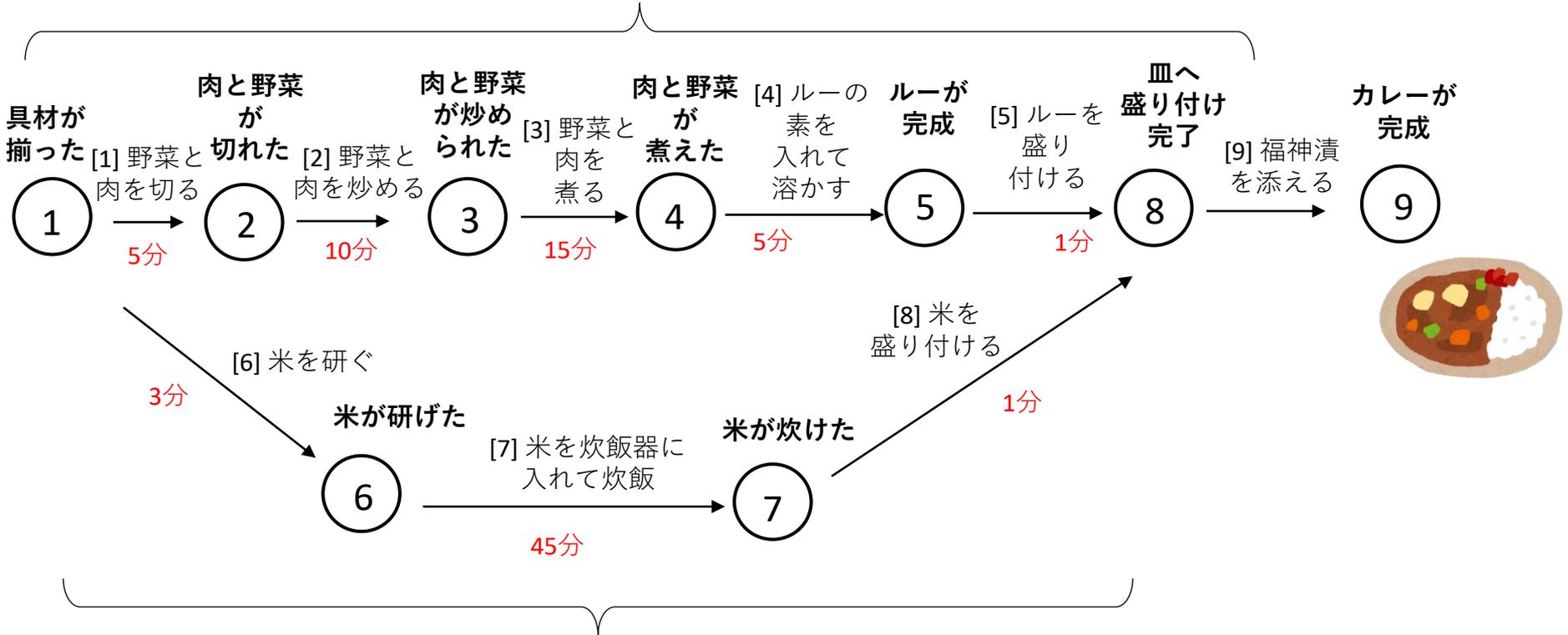
カレー作りのPERT図

(Program Evaluation and Review Technique)



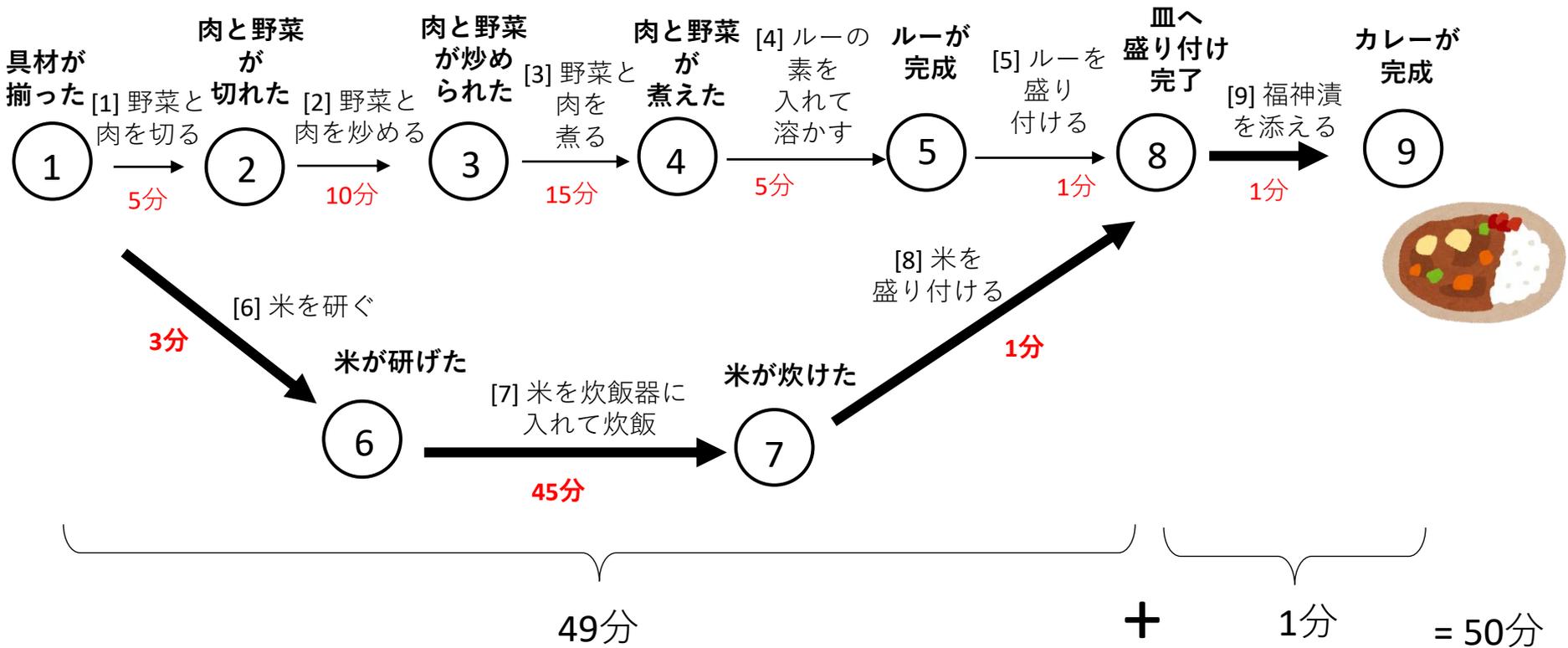
カレー作りに何分かかる？

36分



49分

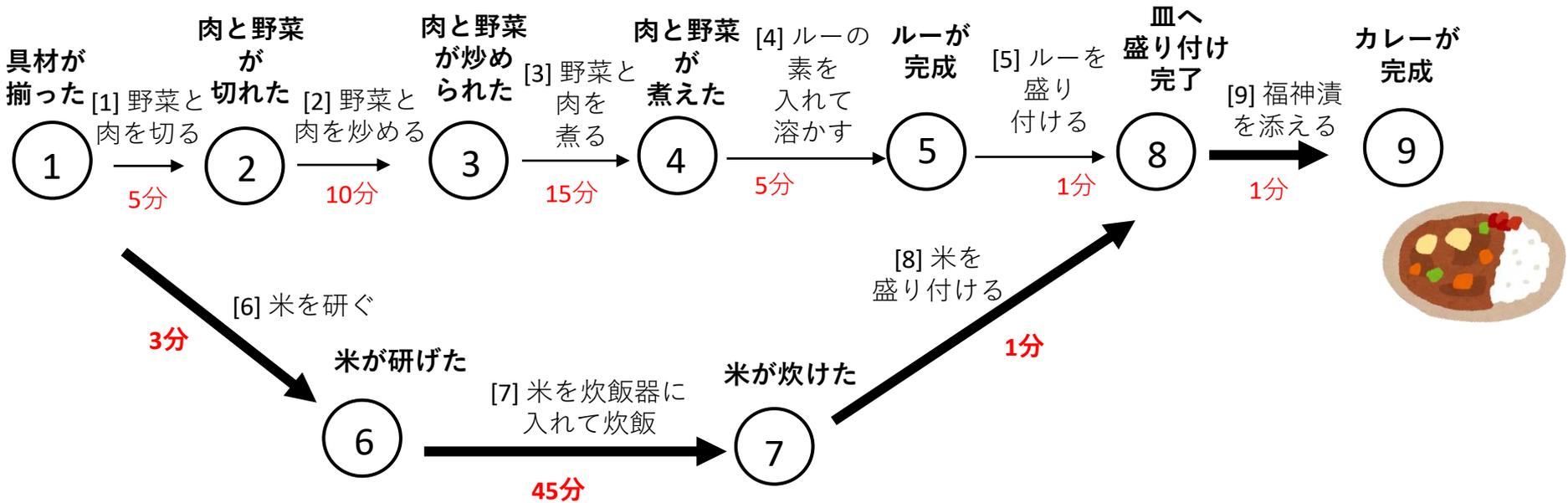
クリティカルパス



かかる時間は50分！

最早結合点時刻

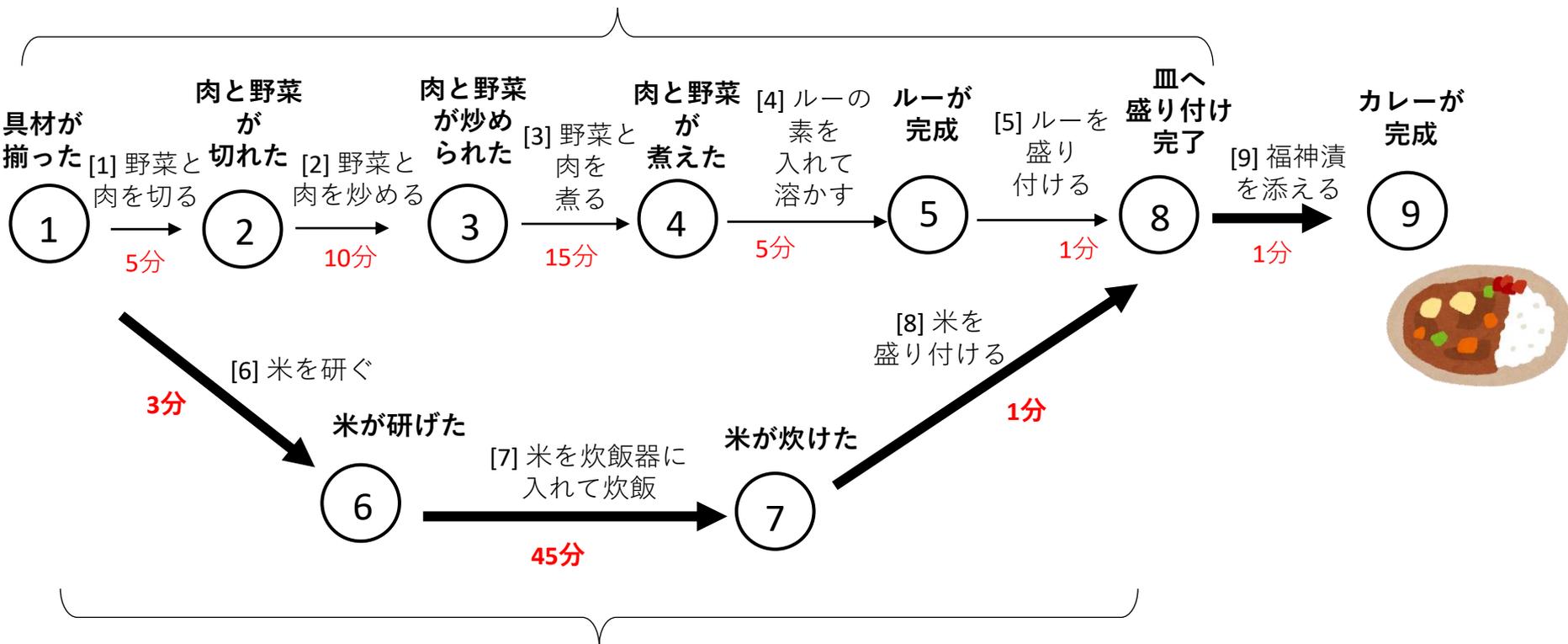
もっとも早く福神漬を添えられるのは何時か？



最早結合点時刻

もっとも早く福神漬を添えられるのは何時か？

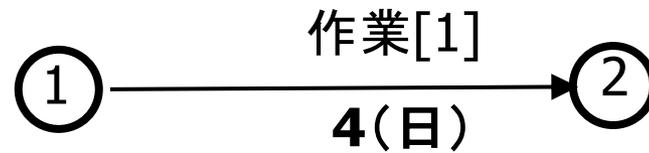
36分



⑧の再送結合点時刻は49分後

2.5 プロジェクトスケジューリング

アローダイアグラム：矢印とノードを使った図示。

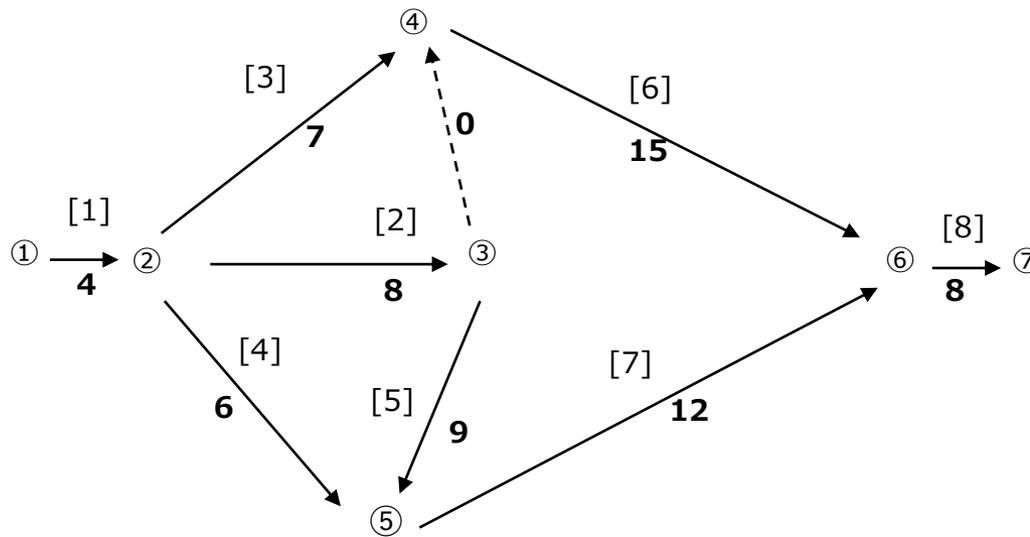


○の中に書かれた数字は作業が終わった状態を表す。

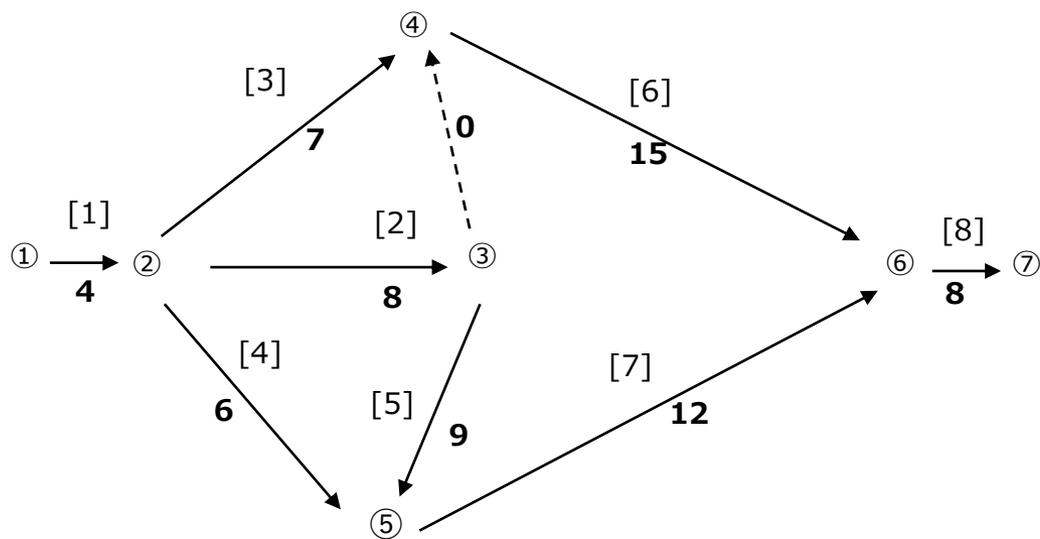
例題：教科書 表 2. 1 3

作業番号	所要時間 (日)	先行作業
[1]	4	なし
[2]	8	[1]
[3]	7	[1]
[4]	6	[1]
[5]	9	[2]
[6]	15	[2],[3]
[7]	12	[4],[5]
[8]	8	[6],[7]

PERT

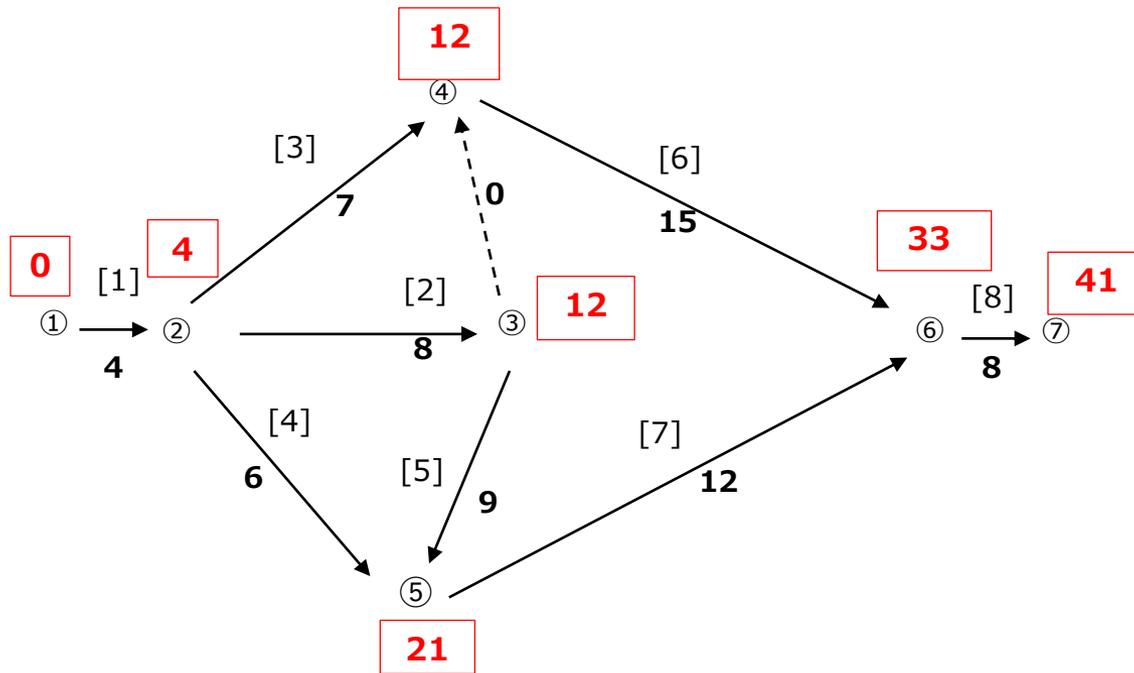


PERT



③→④に相当する作業は無いが、[6]を始めるには[2]が完了していないといけない（③はダミーノード）。

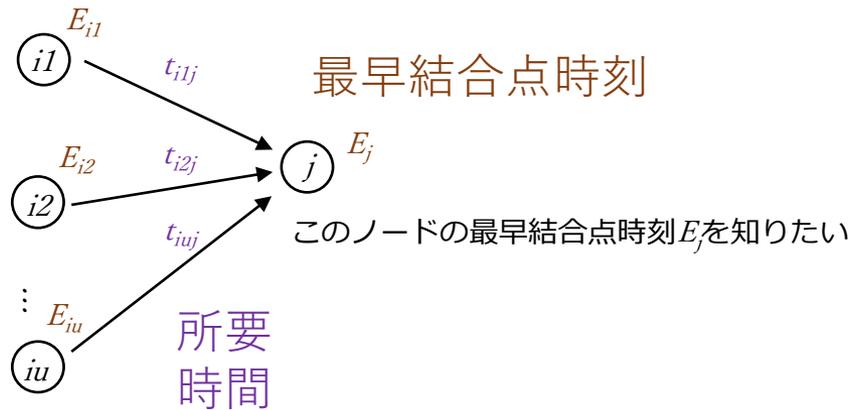
最早結合点時刻(Earliest node time)



ノードに向かう複数の矢印のうち、最も遅い経路にかかる時間

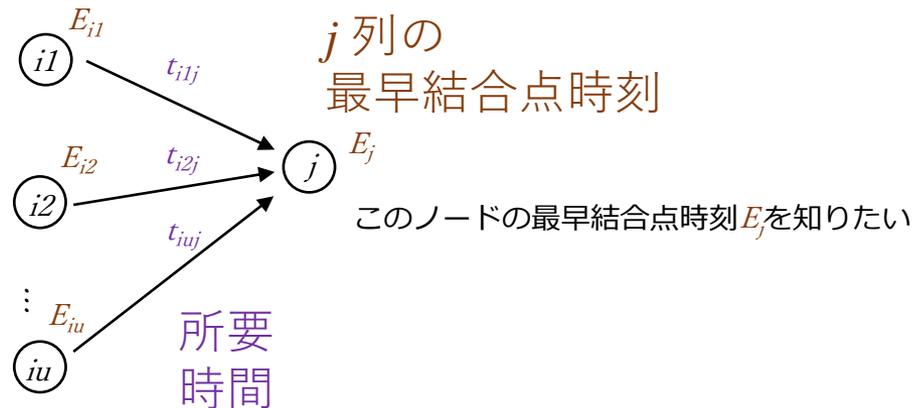
最早結合点時刻を一般化する

最早結合点時刻



最早結合点時刻を一般化する

i 列の
最早結合点時刻



一番遅くまでかかる作業を見つける

$$E_j = \max \{ E_{i1} + t_{i1j}, E_{i2} + t_{i2j}, \dots, E_{iu} + t_{iuj} \}$$

ただし $E_1 = 0$

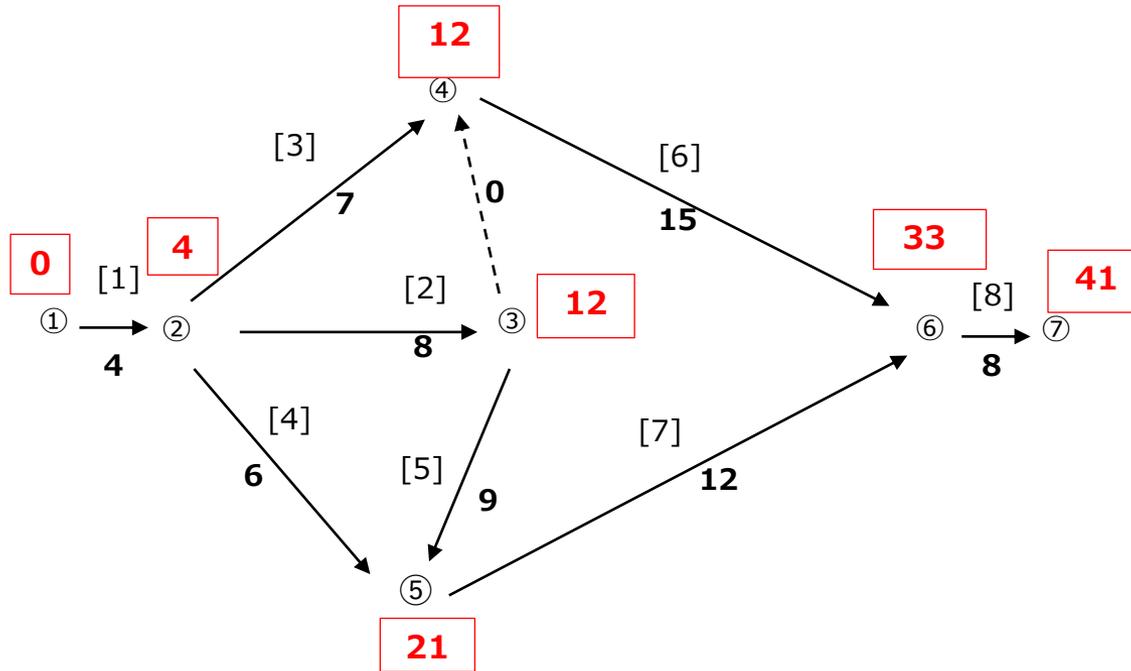
max関数

$$\max\{A, B, C, D, \dots\}$$

A, B, C, Dの中から最も大きい値を選び出す関数

最早開始時刻 (Earliest starting time)

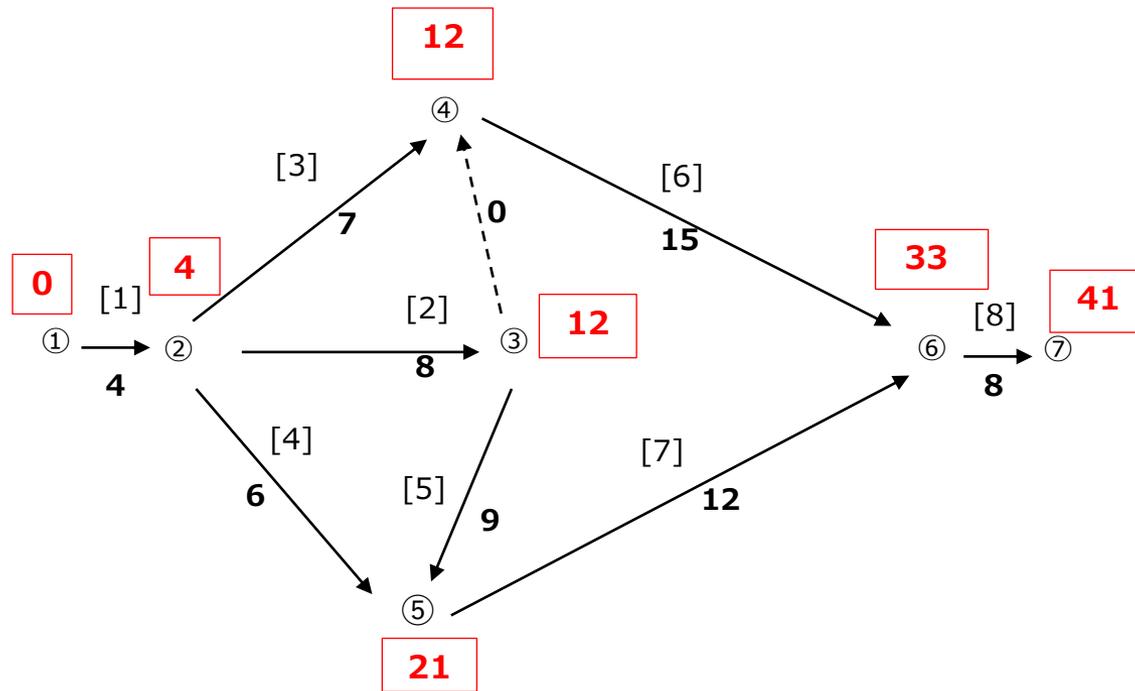
今度はノードではなく各作業（矢印）に着目



例

- 作業[6]の最早開始時刻は12（日）
- 作業[8]の最早開始時刻は33（日）

最早開始時刻を一般化する

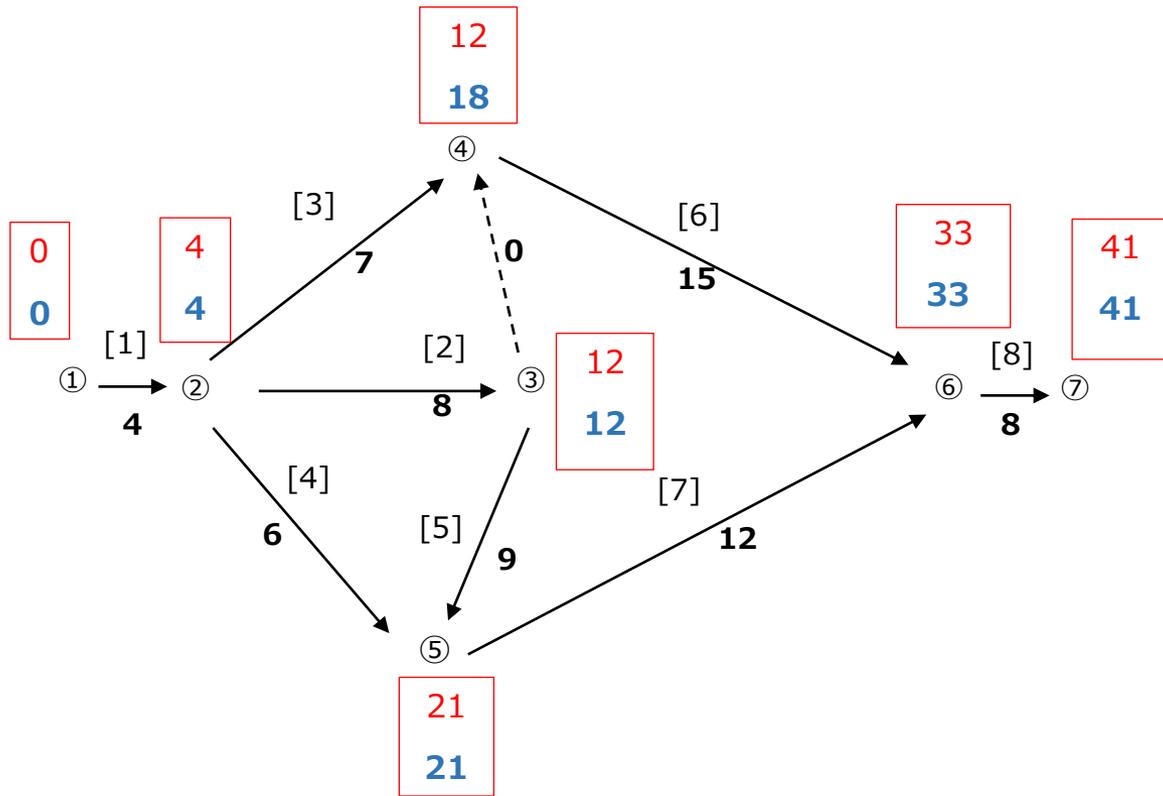


ノード i から j へ向かう作業の最早開始時刻 ES_{ij} は、

$$ES_{ij} = E_i \text{ と求められる}$$

(最早結合点時刻)

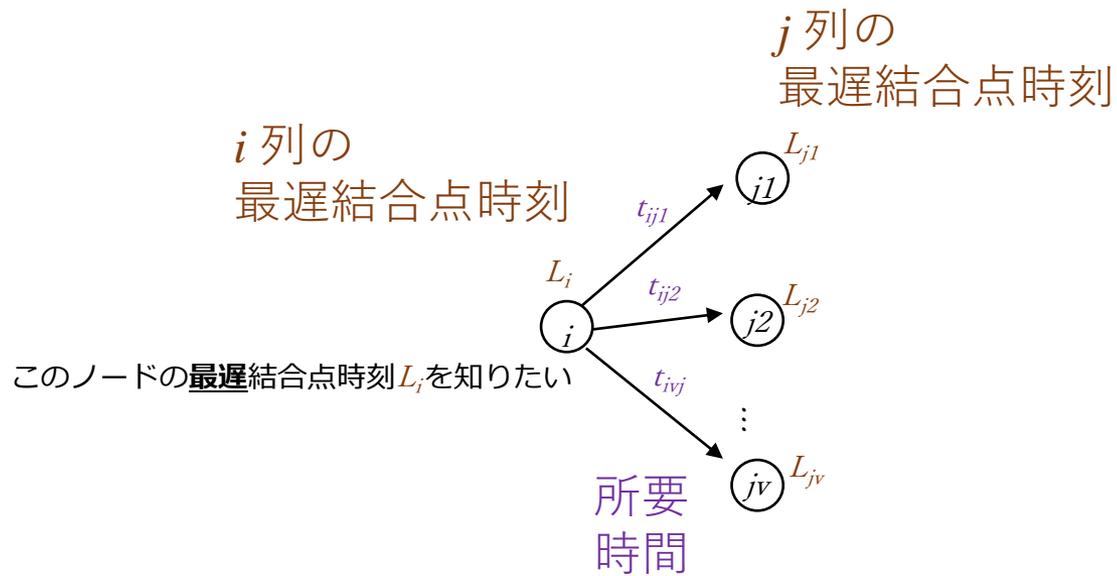
最遅結合点時刻(Latest node time)



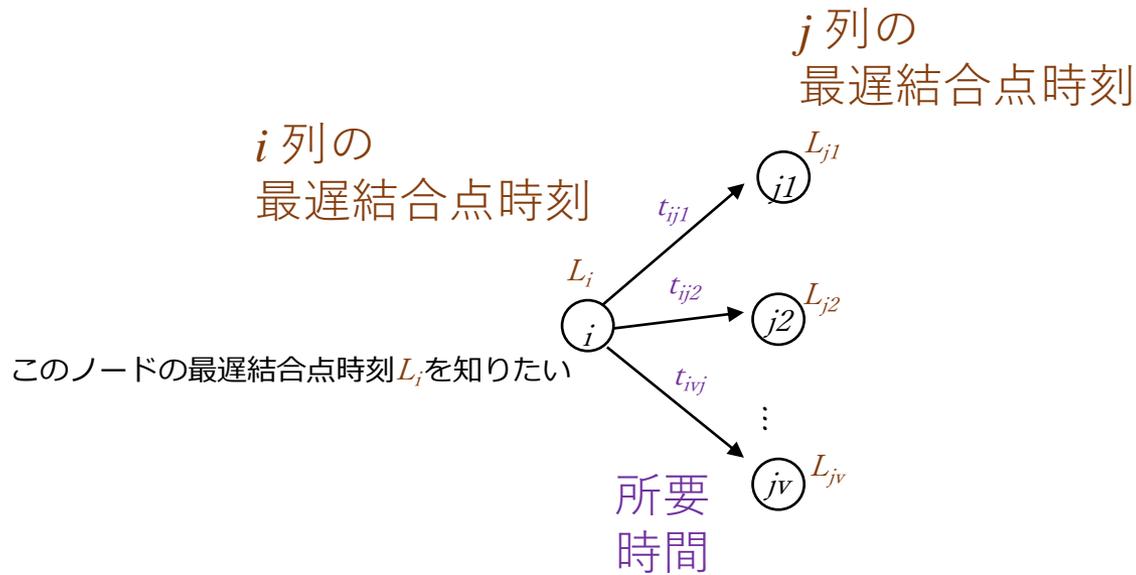
ノードを最後からたどっていく。

全工程を41日で終わらせるためには、最も遅くても何日目までに到達すればよいか？

最遅結合点時刻を一般化する



最遅結合点時刻を一般化する



$$L_i = \min \{ L_{j_1} - t_{ij1}, L_{j_2} - t_{ij2}, \dots, L_{j_v} - t_{ijv} \}$$

ただし $L_n = E_n$
(最終ノードの終了時刻)

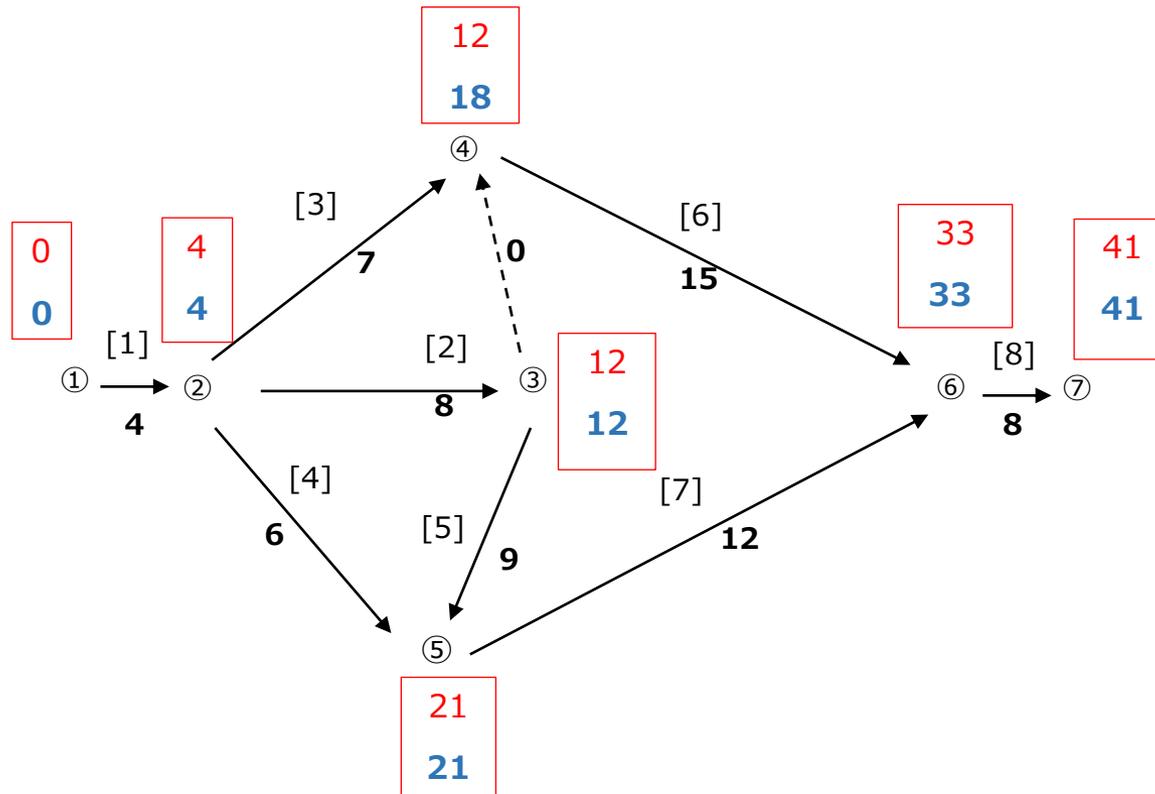
min関数

$$\min\{A, B, C, D, \dots\}$$

A, B, C, Dの中から最も小さい値を選び出す関数

最遅終了時刻 (Latest finishing time)

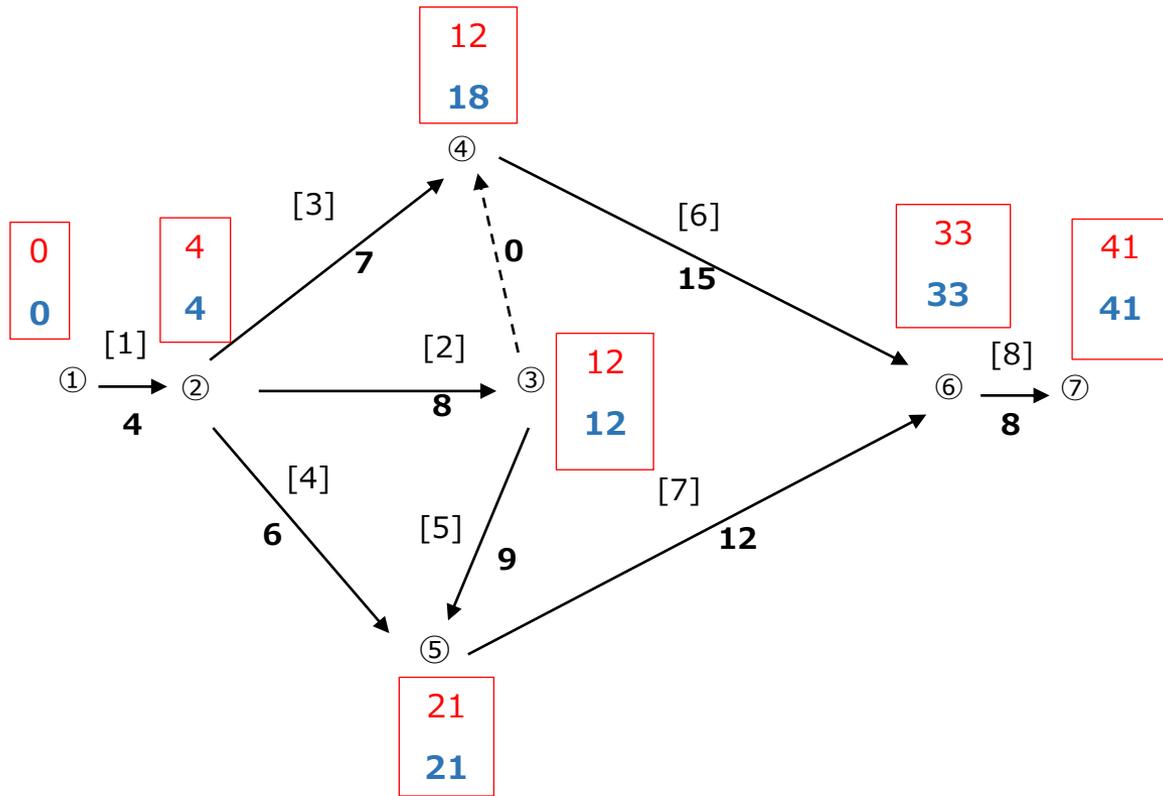
今度はノードではなく各作業（矢印）に着目



例

- 作業[3]の最遅終了時刻は18（日）
- 作業[6]の最遅終了時刻は33（日）

最遅終了時刻を一般化する

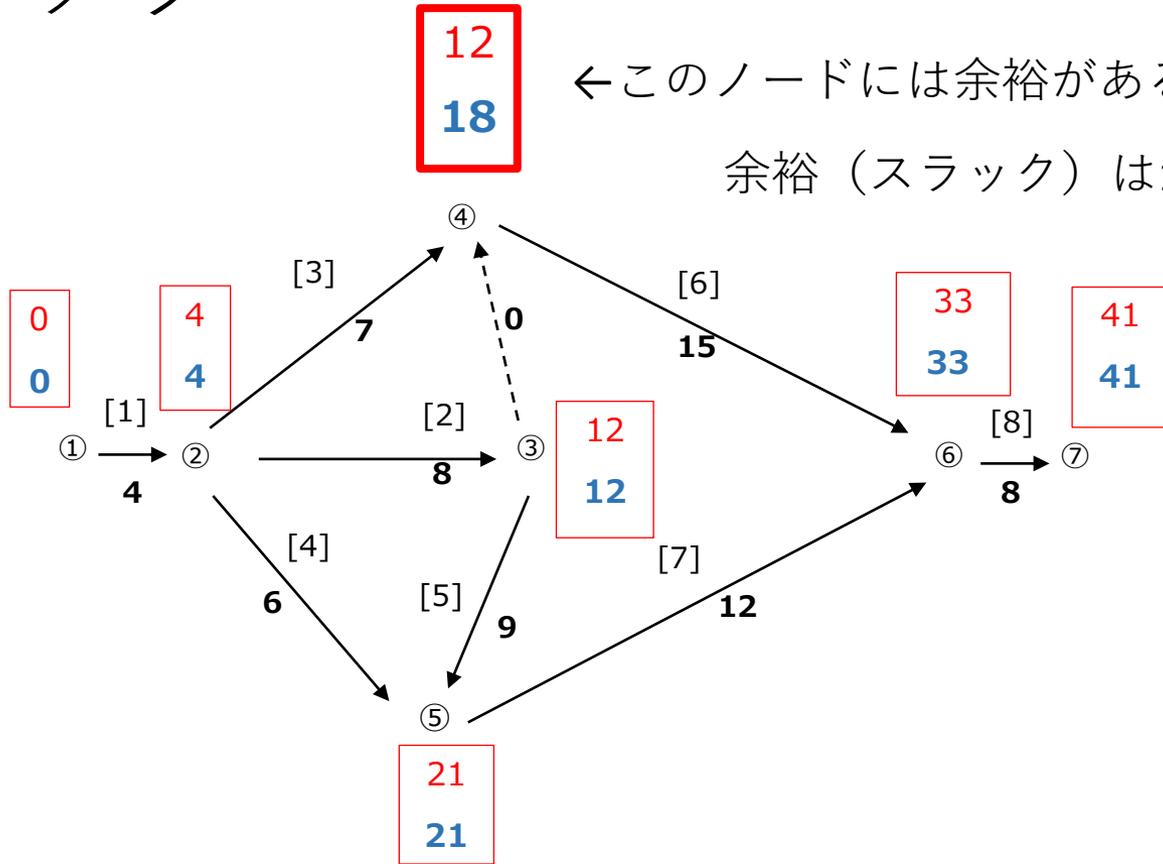


ノード i から j へ向かう作業の最遅終了時刻 LF_{ij} は、

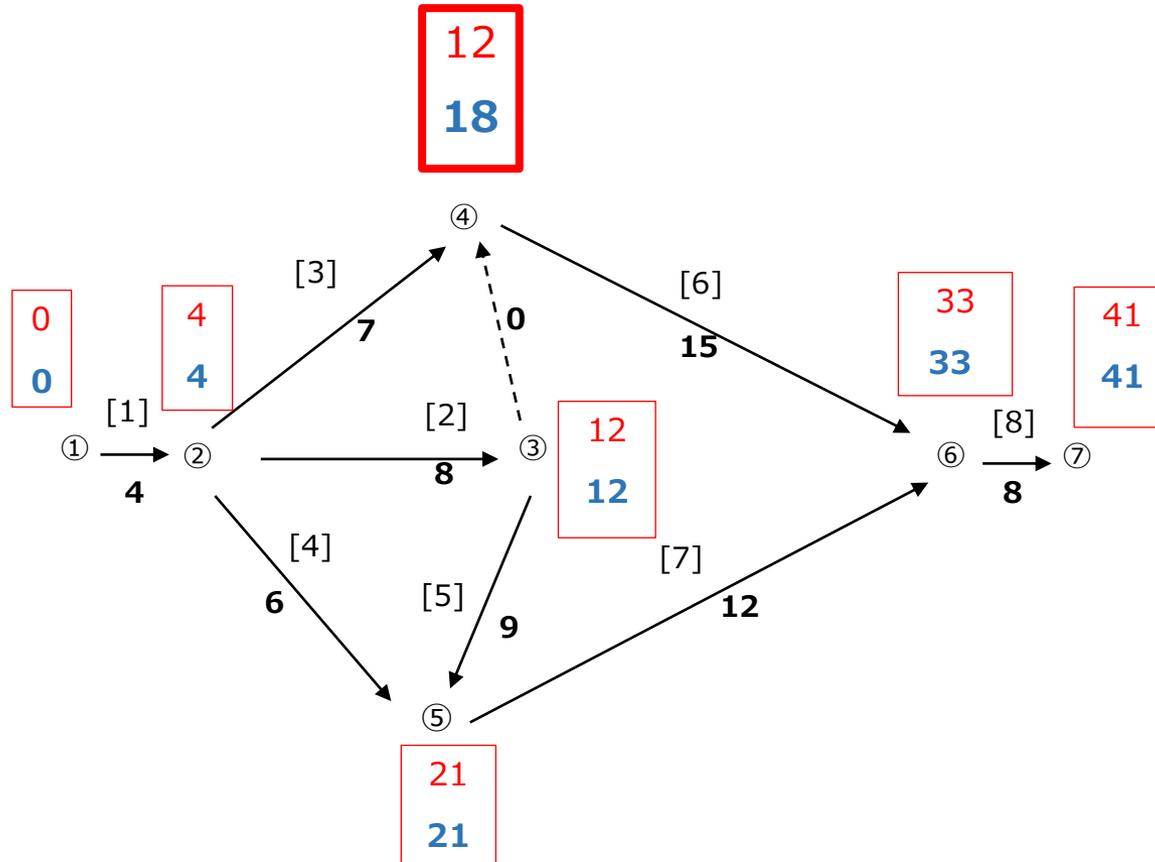
$$LF_{ij} = L_j \text{ と求められる}$$

(最遅結合点時刻)

スラック



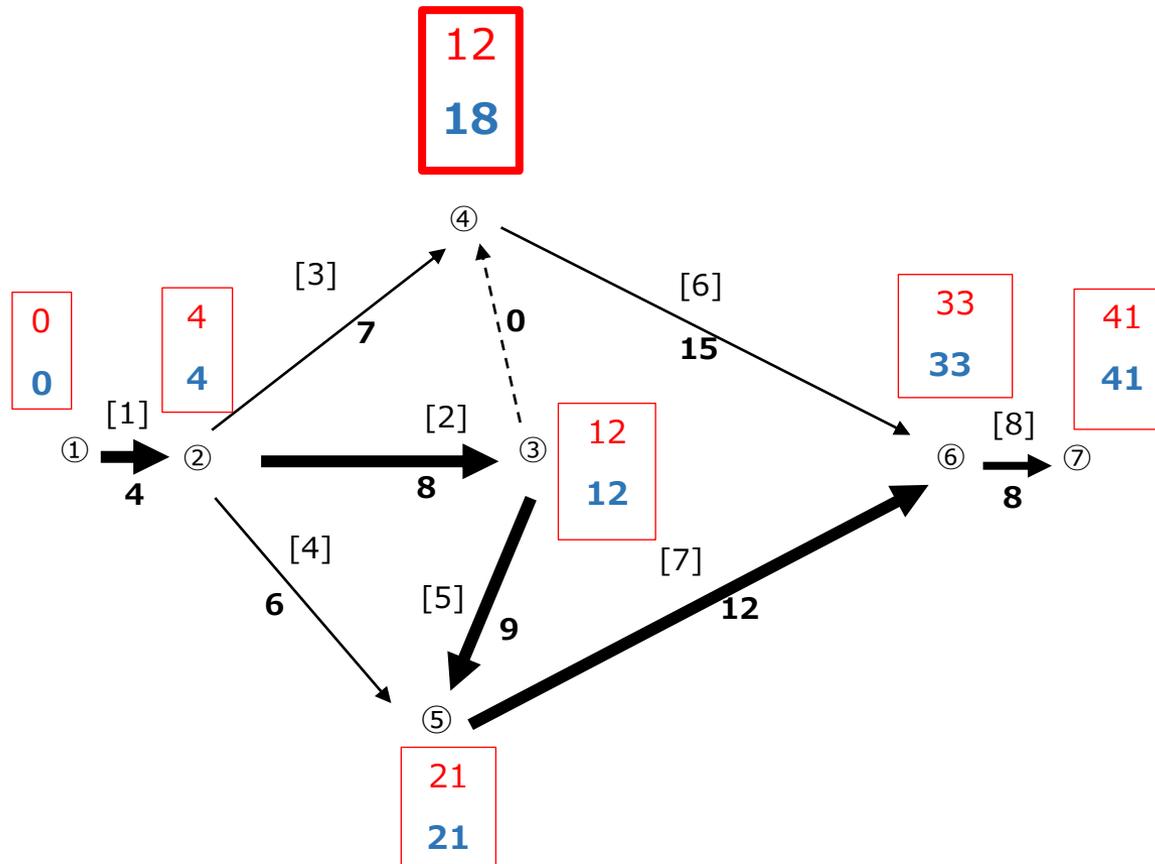
スラックを一般化する



ノード i のスラックは $S_i = L_i - E_i$

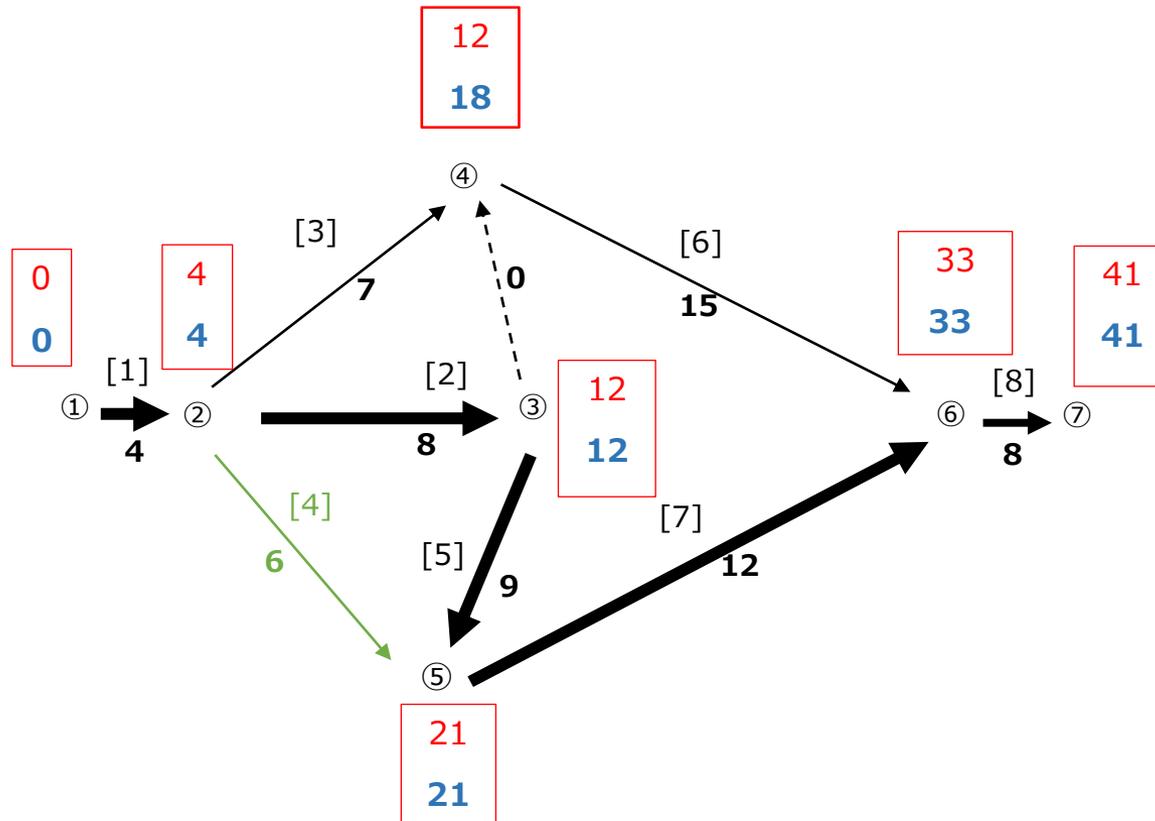
最遅結合点 最早結合点
時刻 時刻

クリティカルパス



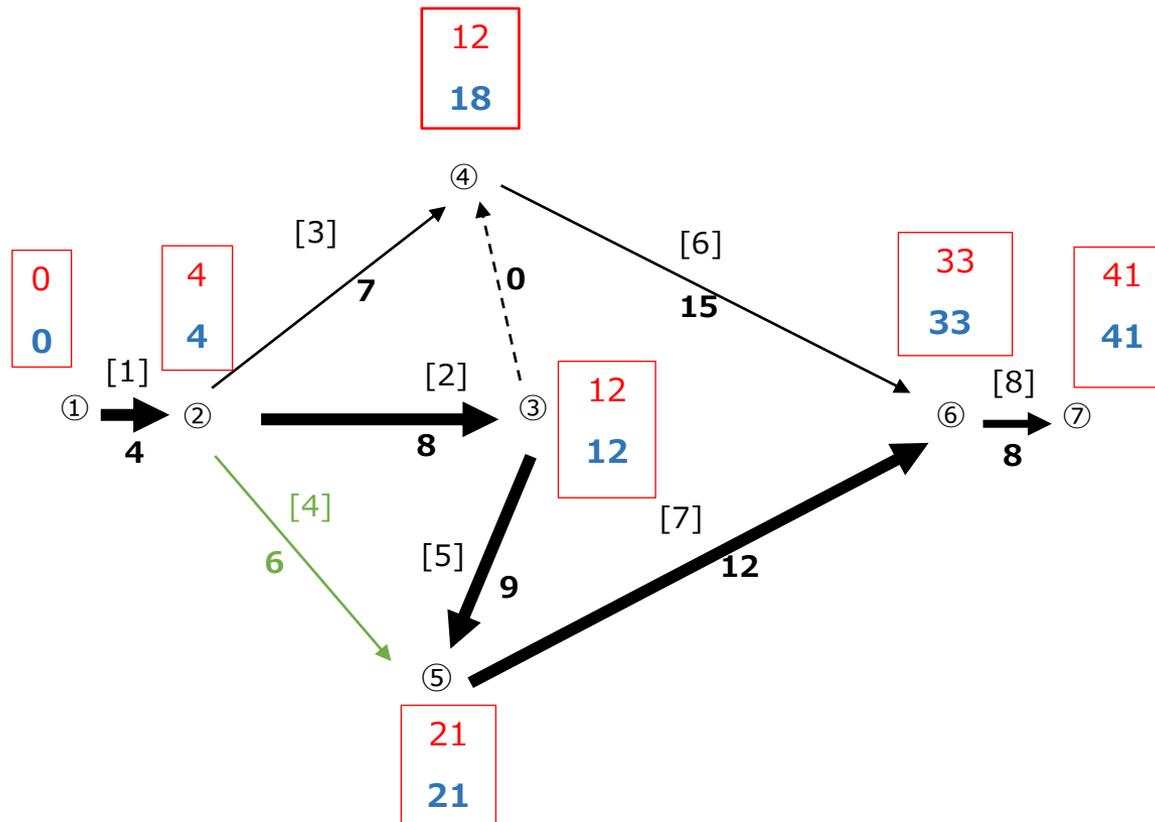
スラックがゼロのノードを全てつないだ経路

フロート



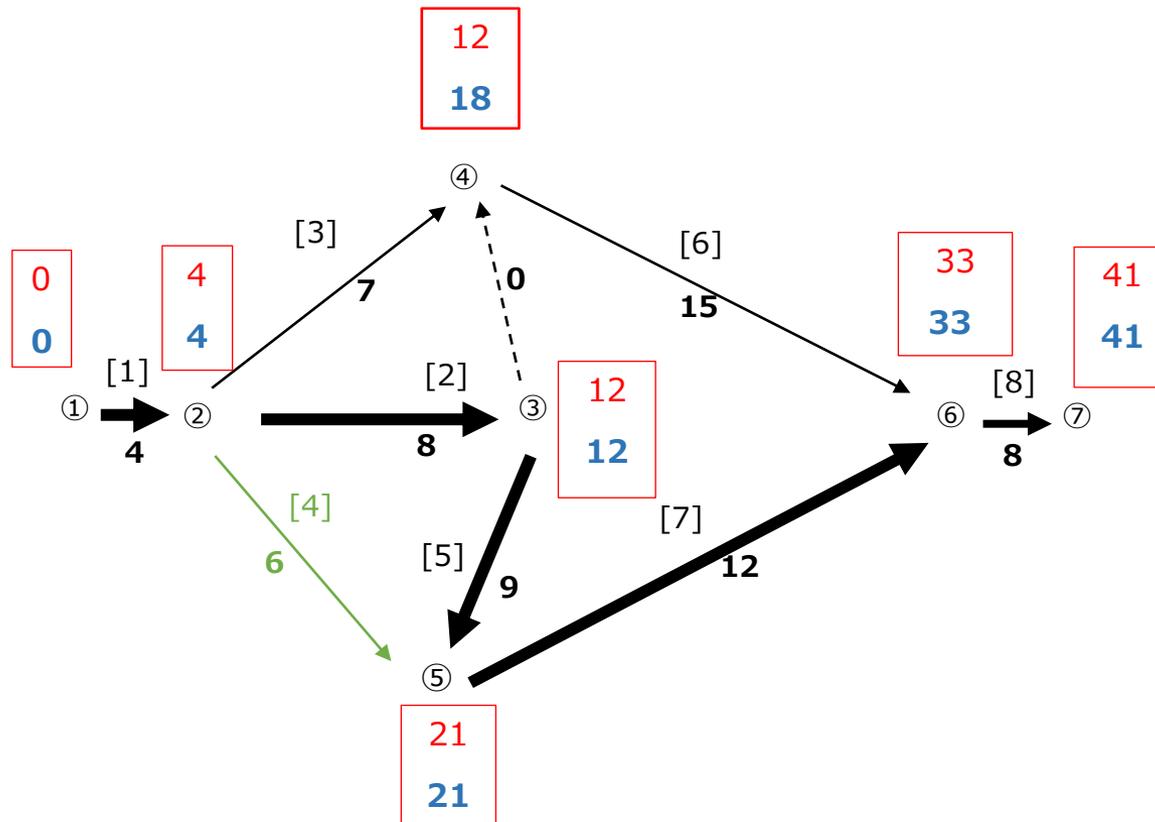
クリティカルパス上にない作業は、ある程度遅延してもプロジェクト全体の工期に影響を及ぼさない。

プロジェクト



[4]の作業はどれくらい遅れても良いだろうか？

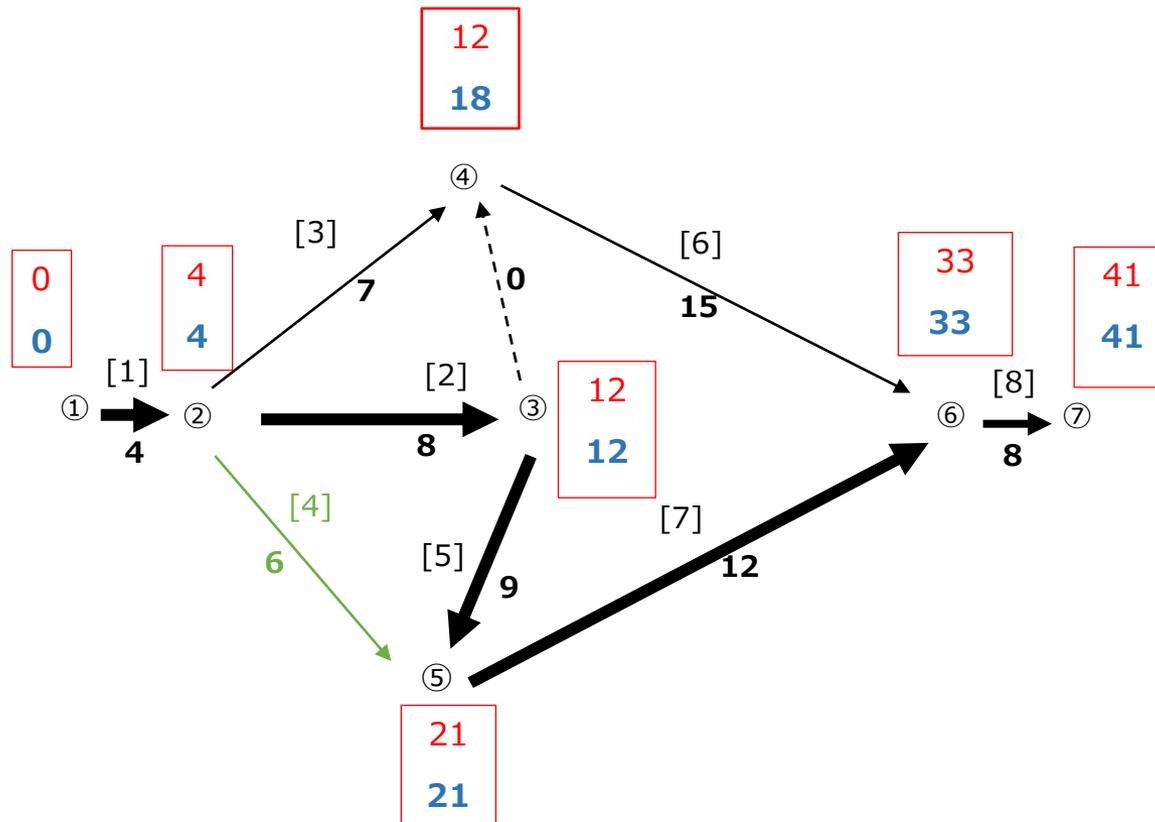
フロート



[4]の作業はどれくらい遅れても良いだろうか？

$21 - 4 - 6 = 11$ 日遅れても良い（フロートと呼ぶ）

フロートを一般化する



ノード*i*からノード*j*の間のフロートを S_{ij} とすると、フロートは

$$S_{ij} = L_j - E_i - t_{ij}$$

最遅結合点 最早結合点 作業時間
時刻 時刻