Cammini minimi

Indice

- Problemi di cammino minimo
  - Cicli negativi
  - Ottimalità dei sottocammini
- Condizioni di ottimalità e potenziali
- La formulazione di programmazione lineare
  - Il modello duale e i potenziali
  - Il modello a sorgente singola
- Cammini minimi nei grafi aciclici
- L'algoritmo di Bellman–Ford
  - Bellman–Ford sul grafo di riferimento
- L'algoritmo di Dijkstra
  - L'algoritmo di Dijkstra sul grafo di riferimento
  - Lunghezze negative
- Errori tipici
- In sintesi

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- enunciare il problema del cammino minimo s–t e quello a sorgente singola, e riconoscere quando un ciclo negativo li rende mal posti;
- usare l'ottimalità dei sottocammini, le equazioni di Bellman e l'albero dei cammini minimi;
- scrivere il modello LP di un cammino minimo come flusso unitario, scriverne il duale con i potenziali π_i e usare potenziali ammissibili come certificato di ottimalità;
- eseguire l'algoritmo per i grafi aciclici, l'algoritmo di Bellman–Ford (con il rilevamento di un ciclo negativo) e l'algoritmo di Dijkstra, dimostrarne la correttezza e darne il tempo di esecuzione.

Prerequisiti

- Dal capitolo «Grafi»: passeggiate, cammini e cicli orientati, la matrice di incidenza nodo-arco, gli alberi e il grafo di riferimento.
- Dal modulo Programmazione lineare, il capitolo «Teoria della dualità»: il duale di un modello LP e il teorema della dualità debole.

Problemi di cammino minimo

Introduzione: Problemi di cammino minimo

- Ogni arco di un grafo orientato riceve una **lunghezza**, e cerchiamo i cammini orientati di lunghezza minima che partono da un nodo sorgente
- I cammini minimi sono ovunque: navigatori stradali, instradamento nelle reti di comunicazione, e come mattone di molti algoritmi sui flussi più avanti nel corso
- Le lunghezze possono essere negative: vedremo quando questo rende il problema mal posto (cicli negativi) e quali proprietà hanno le distanze quando non lo è

Definizione 1: problemi di cammino minimo

Sia $G = (N, A)$ un grafo orientato con una ***lunghezza*** *$d_{ij} \in \mathbb{R}$ per ogni arco $(i, j) \in A$, di qualsiasi segno.*

- *La lunghezza di una passeggiata orientata $W = i_0, i_1, \ldots, i_k$ è la somma delle lunghezze dei suoi archi, $d(W) = \sum_{t=1}^{k} d_{i_{t-1} i_t}$.*
- *Dati due nodi s e t, il* ***problema del cammino minimo*** *s–t chiede un cammino orientato da s a t di lunghezza minima.*
- *Dato un nodo s, la* ***sorgente****, il* ***problema dei cammini minimi a sorgente singola*** *chiede un cammino orientato minimo da s a ogni nodo $j \in N$.*

La lunghezza minima di un cammino orientato da s a j è la ***distanza*** *$\tilde{d}(j)$ di j; poniamo $\tilde{d}(j) = +\infty$ se j non è raggiungibile da s.*

Notazione

- In questo capitolo scriviamo

$$n = |N|, \qquad m = |A|,$$

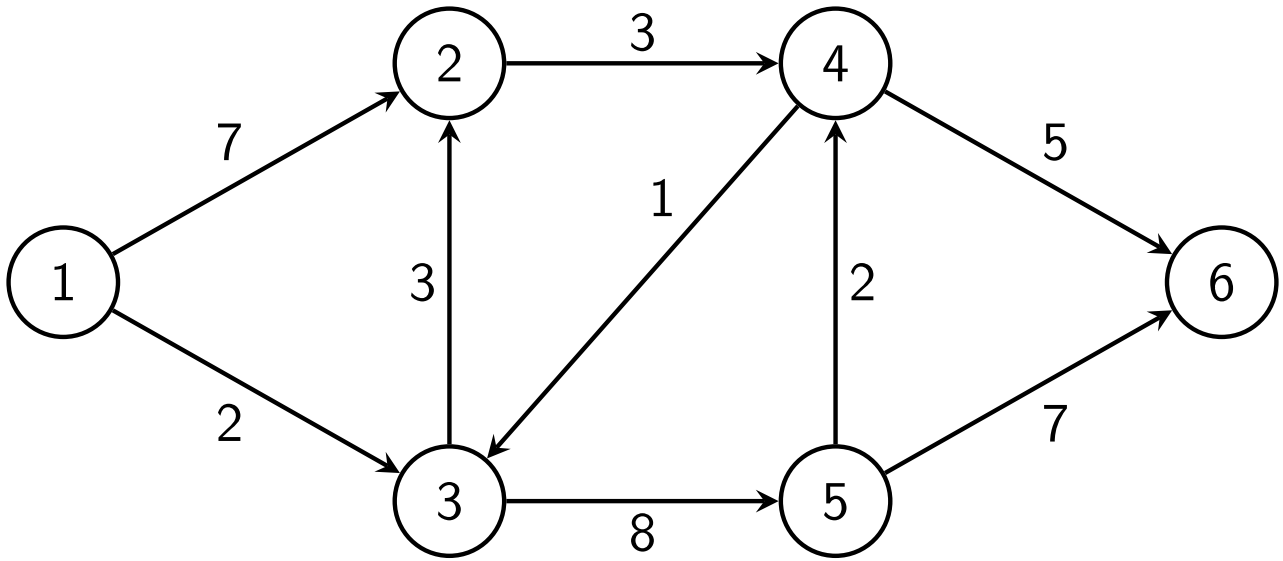
  e "cammino" e "ciclo" indicano sempre un cammino orientato e un ciclo orientato.

- Un algoritmo per il problema a sorgente singola risolve anche il problema s–t, per tutti i t insieme.

- Diamo ora le lunghezze al grafo di riferimento del corso: lo useremo in tutti gli esempi del capitolo.

Esempio (1/2)

Diamo al grafo di riferimento del capitolo «Grafi» le lunghezze d_{ij} scritte sugli archi, tutte non negative; la sorgente è $s = 1$.



Esempio (2/2)

- Tre dei cammini da 1 a 6 e le loro lunghezze sono:

$$d(1,2,4,6) = 7 + 3 + 5 = 15,$$

$$d(1,3,2,4,6) = 2 + 3 + 3 + 5 = 13,$$

$$d(1,3,5,6) = 2 + 8 + 7 = 17.$$

- Le distanze dal nodo 1, calcolate più avanti con l'algoritmo di Dijkstra, sono

$$\tilde{d} = (0,\ 5,\ 2,\ 8,\ 10,\ 13),$$

quindi $1, 3, 2, 4, 6$ è un cammino minimo da 1 a 6.

Le lunghezze possono essere negative.

Che cosa può andare storto? Un ciclo i cui archi hanno lunghezza totale negativa.

Definizione 2: ciclo negativo

*Un **ciclo negativo** è un ciclo orientato C con $d(C) < 0$. Nel resto del capitolo facciamo riferimento a due ipotesi:*

(A1) *ogni nodo di G è raggiungibile da s;*

(A2) *nessun ciclo negativo di G è raggiungibile da s.*

Lemma 1: passeggiate e cammini

Sotto (A2), per ogni passeggiata orientata W da s a j esiste un cammino orientato P da s a j, con archi di W, tale che $d(P) \leq d(W)$.

Dimostrazione

Per induzione sul numero di archi di W. Se W ha nodi distinti, $P = W$.

1. Altrimenti W ripete un nodo: $W = s, \ldots, i_a, \ldots, i_b, \ldots, j$ con $i_a = i_b$, $a < b$ e $b - a$ minimo. Il tratto $C = i_a, \ldots, i_b$ è un ciclo, raggiungibile da s lungo W.

2. Per (A2), $d(C) \geq 0$. Togliendo C da W si ottiene una passeggiata W' da s a j con meno archi e

$$d(W') = d(W) - d(C) \leq d(W).$$

3. Per induzione W' contiene un cammino P con

$$d(P) \leq d(W') \leq d(W).$$

□

Passeggiate e cammini

- Sotto (A2) la lunghezza minima di una passeggiata da s a j coincide con la distanza $\tilde{d}(j)$: gli algoritmi di questo capitolo calcolano lunghezze di passeggiate, e il lemma le trasforma in cammini.

- Con un ciclo negativo C raggiungibile da s e dal quale j è raggiungibile, percorrendo C ancora e ancora si ottengono passeggiate da s a j di lunghezza negativa arbitrariamente grande in valore assoluto.

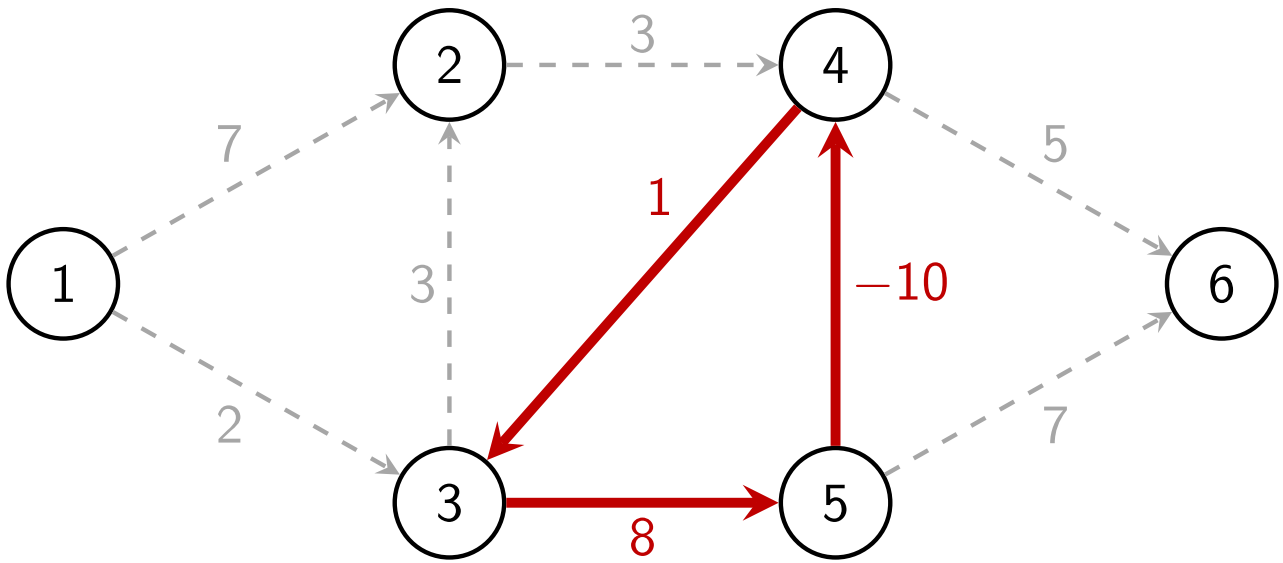
- Gli algoritmi di questo capitolo, allora, rilevano C invece di calcolare le distanze.

Esempio (1/2)

- Il grafo di riferimento ha due cicli, $2, 4, 3, 2$ e $3, 5, 4, 3$. Con le lunghezze del primo esempio le loro lunghezze sono 7 e 11.

- Se la lunghezza dell'arco $(5, 4)$ scende da 2 a $d''_{54} = -10$, il ciclo $3, 5, 4, 3$ (in rosso) ha lunghezza

$$8 - 10 + 1 = -1.$$



Esempio (2/2)

- Le passeggiate da 1 a 6

$$1, 3, 5, 6,$$

$$1, 3, 5, 4, 3, 5, 6,$$

$$1, 3, 5, 4, 3, 5, 4, 3, 5, 6, \ \ldots$$

  hanno lunghezze $17, 16, 15, \ldots$: nessuna passeggiata è minima.

- Useremo di nuovo queste lunghezze d'' con l'algoritmo di Bellman–Ford.

Senza cicli negativi, i cammini minimi hanno una struttura semplice:

ogni tratto di un cammino minimo è a sua volta minimo.

Lemma 2: ottimalità dei sottocammini

Sotto (A2), sia P un cammino minimo da s a j e sia i un nodo di P. Allora il sottocammino P_{si} di P da s a i è un cammino minimo da s a i.

Dimostrazione

Scriviamo P come P_{si} seguito dal sottocammino P_{ij} da i a j, quindi

$$d(P) = d(P_{si}) + d(P_{ij}).$$

Supponiamo per assurdo che un cammino Q da s a i abbia $d(Q) < d(P_{si})$.

1. Q seguito da P_{ij} è una passeggiata da s a j di lunghezza

$$d(Q) + d(P_{ij}) < d(P).$$

2. Per il lemma su passeggiate e cammini essa contiene un cammino da s a j più corto di P: assurdo.

□

Applicata all'ultimo arco di un cammino minimo, l'ottimalità dei sottocammini
dà un sistema di equazioni soddisfatto dalle distanze.

Teorema 1: equazioni di Bellman

Sotto (A1) e (A2), le distanze soddisfano

$$\tilde{d}(s) = 0, \qquad \tilde{d}(j) = \min_{(i,j)\in\delta^-(j)} \left\{ \tilde{d}(i) + d_{ij} \right\}, \quad \forall j \in N \setminus \{s\},$$

e $\tilde{d}(j) \leq \tilde{d}(i) + d_{ij}$ *per ogni arco* $(i,j) \in A$.

Dimostrazione (1/2)

- Tutte le distanze sono finite per (A1).
- $\tilde{d}(s) = 0$: il cammino s ha lunghezza 0, e un cammino da s a s con archi sarebbe un ciclo, di lunghezza ≥ 0 per (A2).

Dimostrazione (2/2)

- *Disuguaglianza.* Un cammino minimo da s a i seguito dall'arco (i, j) è una passeggiata di lunghezza $\tilde{d}(i) + d_{ij}$; per il lemma su passeggiate e cammini,

$$\tilde{d}(j) \leq \tilde{d}(i) + d_{ij}.$$

- *Equazione.* Per $j \neq s$, sia P un cammino minimo da s a j e (i, j) il suo ultimo arco. Per l'ottimalità dei sottocammini,

$$\tilde{d}(j) = d(P) = \tilde{d}(i) + d_{ij},$$

quindi il minimo è raggiunto.

□

Proposizione 1: albero dei cammini minimi

*Sotto (A1) e (A2), esiste un albero ricoprente T di G, l'**albero dei cammini minimi**, tale che per ogni $j \in N$ il cammino da s a j in T è orientato ed è un cammino minimo.*

- *Idea della dimostrazione:* per ogni nodo teniamo l'ultimo arco di un cammino minimo con il minor numero di archi, e mostriamo che questi archi formano un albero.

Dimostrazione (1/2)

Per ogni j, sia $k(j)$ il numero minimo di archi di un cammino minimo da s a j; $k(s) = 0$.

1. Per $j \neq s$, scegliamo un cammino minimo P da s a j con $k(j)$ archi, e sia $pred(j)$ il nodo che precede j su P.

2. Il sottocammino di P da s a $pred(j)$ è minimo (ottimalità dei sottocammini) e ha $k(j) - 1$ archi, quindi

$$k(pred(j)) \leq k(j) - 1 \qquad \text{e} \qquad \tilde{d}(j) = \tilde{d}(pred(j)) + d_{pred(j)\,j}.$$

Dimostrazione (2/2)

3. Passando da j a $pred(j)$ il numero k diminuisce strettamente, quindi la sequenza termina in s. Gli archi $(pred(j), j)$ sono $n-1$ e collegano ogni nodo a s: formano un albero ricoprente (capitolo «Grafi», Teorema «caratterizzazioni di un albero»).

4. Lungo il cammino dell'albero $s = i_0, i_1, \ldots, i_r = j$ ogni arco soddisfa
$$\tilde{d}(i_t) = \tilde{d}(i_{t-1}) + d_{i_{t-1} i_t};$$
sommando, la sua lunghezza è $\tilde{d}(j)$.

□

Esempio (1/2)

Le equazioni di Bellman sul grafo di riferimento, con le distanze $\tilde{d} = (0, 5, 2, 8, 10, 13)$ del primo esempio:

- il nodo 4 ha gli archi entranti $(2, 4)$ e $(5, 4)$, e

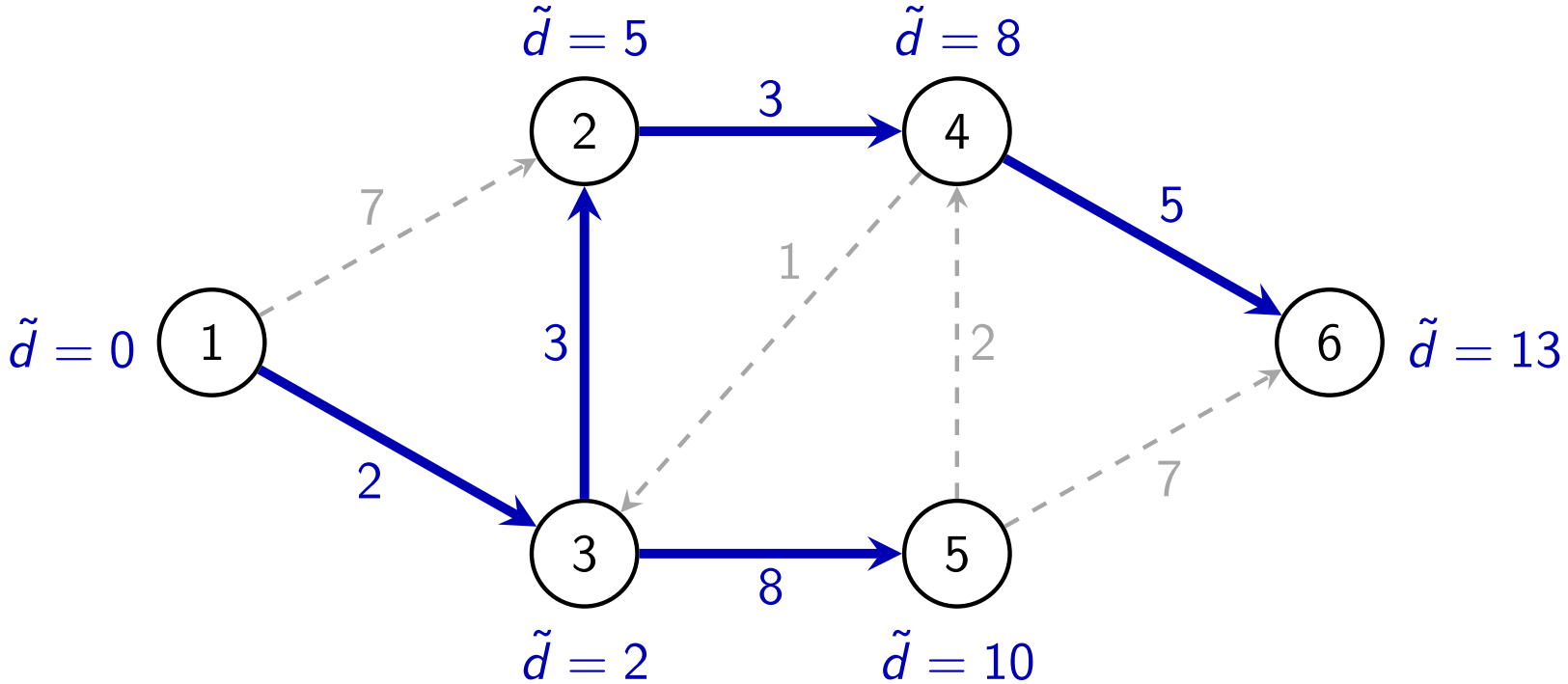
$$\tilde{d}(4) = \min\{5 + 3,\ 10 + 2\} = 8;$$

- il nodo 6 ha gli archi entranti $(4, 6)$ e $(5, 6)$, e

$$\tilde{d}(6) = \min\{8 + 5,\ 10 + 7\} = 13.$$

Esempio (2/2)

Gli archi che realizzano il minimo formano l'albero dei cammini minimi $(1,3)$, $(3,2)$, $(2,4)$, $(3,5)$, $(4,6)$, in blu; accanto a ogni nodo la sua distanza.



Riepilogo

Problemi di cammino minimo:

- problemi s–t e a sorgente singola; la distanza $\tilde{d}(j)$ è la lunghezza di un cammino minimo
- Un ciclo negativo raggiungibile da s rende le passeggiate arbitrariamente corte: ipotesi (A1) e (A2)

Struttura dei cammini minimi:

- Senza cicli negativi raggiungibili dalla sorgente, ogni sottocammino di un cammino minimo è minimo
- Le distanze soddisfano le equazioni di Bellman
- Un solo albero ricoprente contiene un cammino minimo verso ogni nodo

Condizioni di ottimalità e potenziali

Introduzione: Condizioni di ottimalità e potenziali

- Qualcuno ci dà un cammino e sostiene che è minimo: come possiamo verificarlo senza elencare tutti i cammini?
- Diamo un numero, il **potenziale**, a ogni nodo, e guardiamo i **costi ridotti** degli archi
- Potenziali ammissibili sono un **certificato di ottimalità**; scopriremo che sono le variabili duali del modello LP della prossima sezione

Definizione 3: potenziali e costi ridotti

Un vettore $\boldsymbol{\pi} \in \mathbb{R}^n$ *assegna un* ***potenziale*** π_i *a ogni nodo* $i \in N$. *Il* ***costo ridotto*** *dell'arco* (i, j) *rispetto a* $\boldsymbol{\pi}$ *è*

$$\bar{c}_{ij} = d_{ij} - \pi_i + \pi_j.$$

I potenziali sono ***ammissibili*** *se* $\bar{c}_{ij} \geq 0$ *per ogni arco* $(i, j) \in A$, *e un arco con* $\bar{c}_{ij} = 0$ *è* ***stretto***.

Etichette e potenziali

- Date delle **etichette** $d(j)$, lunghezze di passeggiate da s a j, usiamo i potenziali $\pi_i = -d(i)$. Allora

$$\bar{c}_{ij} = d_{ij} + d(i) - d(j),$$

  e i potenziali sono ammissibili se e solo se $d(j) \leq d(i) + d_{ij}$ per ogni arco.

- Lungo una passeggiata $W = i_0, i_1, \ldots, i_k$ i potenziali si elidono a catena (somma telescopica):

$$d(W) = \sum_{t=1}^{k} \bar{c}_{i_{t-1} i_t} \; + \; \pi_{i_0} - \pi_{i_k}.$$

Teorema 2: potenziali come certificato di ottimalità

Siano π potenziali ammissibili e sia P un cammino da s a t i cui archi sono tutti stretti. Allora P è un cammino minimo da s a t, e $d(P) = \pi_s - \pi_t$. Inoltre G non ha cicli negativi.

Dimostrazione

Usiamo la formula telescopica qui sopra.

1. Gli archi di P hanno $\bar{c}_{ij} = 0$, quindi $d(P) = \pi_s - \pi_t$.
2. Per ogni cammino Q da s a t,

$$d(Q) = \sum_{(i,j)\in Q} \bar{c}_{ij} + \pi_s - \pi_t \geq \pi_s - \pi_t = d(P).$$

3. Per ogni ciclo C il primo e l'ultimo nodo coincidono, quindi

$$d(C) = \sum_{(i,j)\in C} \bar{c}_{ij} \geq 0.$$

□

Corollario 1: condizioni di ottimalità per le etichette

Siano le etichette $d(j)$, $j \in N$, finite con $d(s) = 0$, e supponiamo che

(i) $d(j) \leq d(i) + d_{ij}$ *per ogni arco* $(i, j) \in A$;

(ii) *per ogni* $j \neq s$ *esista un cammino da* s *a* j *di lunghezza* $d(j)$.

Allora $d(j) = \tilde{d}(j)$ *per ogni* $j \in N$.

Dimostrazione

- Per (i) i potenziali $\pi_i = -d(i)$ sono ammissibili, quindi ogni cammino Q da s a j ha

$$d(Q) \geq \pi_s - \pi_j = d(j)$$

con il ragionamento del teorema sui potenziali come certificato di ottimalità.

- Per (ii) il valore $d(j)$ è realizzato da un cammino, quindi $d(j) = \tilde{d}(j)$.

□

Esempio (1/2)

Un certificato sul grafo di riferimento: con le distanze del primo esempio i potenziali sono

$$\boldsymbol{\pi} = -\tilde{\boldsymbol{d}} = (0, -5, -2, -8, -10, -13),$$

e i costi ridotti $\bar{c}_{ij} = d_{ij} - \pi_i + \pi_j$ sono:

arco	$(1,2)$	$(1,3)$	$(2,4)$	$(3,2)$	$(3,5)$	$(4,3)$	$(4,6)$	$(5,4)$	$(5,6)$
d_{ij}	7	2	3	3	8	1	5	2	7
$\bar{c}_{ij}$	2	0	0	0	0	7	0	4	4

Esempio (2/2)

- Tutti i costi ridotti sono non negativi, e gli archi stretti sono gli archi dell'albero dei cammini minimi. Il cammino $1, 3, 2, 4, 6$ è stretto, quindi è minimo, con lunghezza $\pi_1 - \pi_6 = 13$.

- L'albero della visita in ampiezza, $(1, 2)$, $(1, 3)$, $(2, 4)$, $(3, 5)$, $(4, 6)$, dà le etichette $(0, 7, 2, 10, 10, 15)$. Con i loro potenziali,

$$\bar{c}_{32} = 3 - (-2) + (-7) = -2 < 0 :$$

  i potenziali non sono ammissibili, ed entrare nel nodo 2 attraverso $(3, 2)$ accorcia il suo cammino di 2.

Riepilogo

Potenziali:

- Costi ridotti $\bar{c}_{ij} = d_{ij} - \pi_i + \pi_j$; i potenziali ammissibili hanno $\bar{c}_{ij} \geq 0$ su ogni arco
- Potenziali ammissibili e un cammino fatto di archi stretti dimostrano che il cammino è minimo

Etichette:

- Con le etichette, $\pi_i = -d(i)$, e l'ammissibilità si legge $d(j) \leq d(i) + d_{ij}$
- Etichette che la soddisfano e che sono lunghezze di cammini sono le distanze

La formulazione di programmazione lineare

Introduzione: La formulazione di programmazione lineare

- Un cammino da s a t si può vedere come **un'unità di flusso** inviata da s a t
- Si ottiene un modello LP con la matrice di incidenza nodo-arco; il suo duale è proprio il problema dei potenziali ammissibili
- Lo stesso modello di flusso, con altri bilanci e con le capacità, è il punto di partenza dei capitoli sui flussi più avanti nel corso

Un cammino come flusso unitario

- Un cammino da s a t invia un'unità di flusso da s a t.
- Usiamo una variabile $x_{ij} \geq 0$ per ogni arco, il flusso sull'arco, e i **bilanci**

$$b_s = 1, \qquad b_t = -1, \qquad b_i = 0, \quad \forall i \in N \setminus \{s, t\}.$$

Il modello $\mathrm{SP}(s,t)$

Usando queste variabili, un modello LP per il problema del cammino minimo s–t è il seguente:

$$\min \sum_{(i,j)\in A} d_{ij}\, x_{ij}$$

$$\text{soggetto a} \sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(j,i)\in\delta^-(i)} x_{ji} = b_i, \quad \forall i \in N,$$

$$x_{ij} \geq 0, \quad \forall (i,j) \in A.$$

Questo modello LP si indica come modello $\mathrm{SP}(s,t)$.

Descrizione del modello $\mathrm{SP}(s, t)$

Descrizione della funzione obiettivo e dei vincoli del modello $\mathrm{SP}(s, t)$:

- la funzione obiettivo minimizza la lunghezza totale del flusso;
- i vincoli di uguaglianza sono i vincoli di conservazione del flusso: un'unità esce da s, un'unità entra in t, e ogni altro nodo invia quello che riceve;
- infine, i vincoli di non negatività definiscono le variabili del modello LP.

Forma matriciale e vettori di incidenza

- In forma matriciale il modello è

$$\min\{\boldsymbol{d}'\boldsymbol{x} : \boldsymbol{B}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq 0\},$$

  con la matrice di incidenza nodo-arco $\boldsymbol{B}$ del capitolo «Grafi»: $+1$ nella coda e -1 nella testa di ogni arco.

- Il **vettore di incidenza** $\boldsymbol{x}^P$ di un cammino P da s a t, con $x^P_{ij} = 1$ sugli archi di P e 0 altrove, è ammissibile, e il suo valore è $d(P)$.

Le variabili duali

- Il vincolo del nodo i è un'equazione, quindi la sua variabile duale π_i è libera.
- La colonna di x_{ij} in $\boldsymbol{B}$ ha $+1$ nella riga i e -1 nella riga j, quindi il vincolo duale di $x_{ij} \geq 0$ è

$$\pi_i - \pi_j \leq d_{ij}$$

(modulo Programmazione lineare, capitolo «Teoria della dualità»).

Il modello $D(\mathrm{SP}(s,t))$

Usando una variabile duale libera π_i per ciascun vincolo di conservazione del flusso, il modello duale del modello $\mathrm{SP}(s,t)$ è:

$$\begin{aligned} \max \quad & \pi_s - \pi_t \\ \text{soggetto a} \quad & \pi_i - \pi_j \leq d_{ij}, \quad \forall (i,j) \in A, \\ & \pi_i \in \mathbb{R}, \quad \forall i \in N. \end{aligned}$$

Questo modello LP si indica come modello $D(\mathrm{SP}(s,t))$.

Soluzioni duali e potenziali

- Il vincolo duale è

$$\bar{c}_{ij} = d_{ij} - \pi_i + \pi_j \geq 0 :$$

  le soluzioni ammissibili del duale sono i potenziali ammissibili della sezione precedente.

- $\bar{c}_{ij}$ è il costo ridotto di x_{ij} nel senso della programmazione lineare.

Teorema 3: LP e cammini minimi

Sotto (A1) e (A2), il modello $\mathrm{SP}(s,t)$ *ha una soluzione ottima e* $z(\mathrm{SP}(s,t)) = \tilde{d}(t)$*. Il vettore di incidenza di un cammino minimo da* s *a* t *è ottimo per* $\mathrm{SP}(s,t)$*, e i potenziali* $\pi_i = -\tilde{d}(i)$ *sono ottimi per* $D(\mathrm{SP}(s,t))$*.*

Dimostrazione

Sia P un cammino minimo da s a t, che esiste per (A1).

1. $\boldsymbol{x}^P$ è ammissibile per $\mathrm{SP}(s,t)$, con valore $d(P) = \tilde{d}(t)$.
2. Per le equazioni di Bellman, $\tilde{d}(j) \leq \tilde{d}(i) + d_{ij}$ per ogni arco, quindi $\pi_i = -\tilde{d}(i)$ soddisfa $\pi_i - \pi_j \leq d_{ij}$; il suo valore è
$$\pi_s - \pi_t = 0 + \tilde{d}(t).$$
3. Una soluzione primale ammissibile e una soluzione duale ammissibile con lo stesso valore sono entrambe ottime, per la dualità debole.

□

Proposizione 2: un ciclo negativo rende il modello illimitato

Se $\mathrm{SP}(s,t)$ *è ammissibile e* G *ha un ciclo negativo* C*, raggiungibile da* s *o no, allora* $\mathrm{SP}(s,t)$ *è illimitato.*

Dimostrazione

- Sia $\bar{\boldsymbol{x}}$ ammissibile e sia $\boldsymbol{x}^C$ il vettore di incidenza di C.
- Le colonne degli archi di C sommano a zero (capitolo «Grafi», Lemma «colonne di un ciclo»), quindi $\boldsymbol{B}\boldsymbol{x}^C = 0$.
- Per ogni $\lambda \geq 0$ il vettore $\bar{\boldsymbol{x}} + \lambda \boldsymbol{x}^C$ è ammissibile, con valore

$$\boldsymbol{d}'\bar{\boldsymbol{x}} + \lambda\, d(C),$$

che tende a $-\infty$.

□

Osservazione 1: Il segno dei potenziali

Con la matrice di incidenza del corso ($+1$ nella coda) il vincolo duale è $\pi_i - \pi_j \leq d_{ij}$, e i potenziali ottimi sono $\pi_i = -\tilde{d}(i)$, a meno di una costante additiva. La forma $\pi_j - \pi_i \leq d_{ij}$, con $\pi_i = \tilde{d}(i)$, appartiene alla convenzione di segno opposta (-1 nella coda).

Il modello a sorgente singola

- Per trovare un cammino minimo da s a ogni nodo, s invia un'unità a ciascuno degli altri $n-1$ nodi:

$$b_s = n-1, \qquad b_i = -1, \quad \forall i \in N \setminus \{s\}.$$

- Con questi bilanci il modello $\mathrm{SP}(s,t)$ si chiama $\mathrm{SP}(s)$.

- Il suo duale massimizza $(n-1)\pi_s - \sum_{i \neq s} \pi_i$ con gli stessi vincoli $\pi_i - \pi_j \leq d_{ij}$.

Proposizione 3: il modello a sorgente singola

Sotto (A1) e (A2), sia T un albero dei cammini minimi e, per ogni $j \neq s$, sia T_j l'insieme dei nodi il cui cammino nell'albero passa per j (j compreso). Il flusso con $x_{ij} = |T_j|$ sugli archi (i,j) di T e 0 altrove è ottimo per $\mathrm{SP}(s)$, i potenziali $\pi_i = -\tilde{d}(i)$ sono ottimi per il suo duale, e

$$z(\mathrm{SP}(s)) = \sum_{j \in N} \tilde{d}(j).$$

Dimostrazione

1. *Ammissibilità.* In un nodo $j \neq s$ il flusso che entra sull'arco dell'albero è $|T_j|$, e il flusso che esce è $\sum |T_h|$ sugli archi dell'albero (j, h), cioè $|T_j| - 1$: il bilancio è -1. In s il flusso uscente copre tutti gli altri nodi: $n - 1$.

2. *Valore.* Il flusso è la somma dei vettori di incidenza degli $n - 1$ cammini dell'albero, perché l'arco (i, j) sta sui cammini dell'albero dei $|T_j|$ nodi di T_j. Il suo valore è $\sum_j \tilde{d}(j)$.

3. *Duale.* $\pi_i = -\tilde{d}(i)$ è ammissibile come nel teorema su LP e cammini minimi, con valore $(n-1) \cdot 0 + \sum_{i \neq s} \tilde{d}(i)$. Per la dualità debole entrambe le soluzioni sono ottime.

□

Vertici interi

- Nel capitolo «Modelli di flusso su reti» si mostra che i vertici di questi poliedri sono interi, per la totale unimodularità di $\boldsymbol{B}$.
- Risolviamo ora i due modelli sul grafo di riferimento.

Esempio (1/3)

Risolviamo i modelli con le lunghezze del primo esempio, $s = 1$ e $t = 6$ (valori verificati con Gurobi).

- $\mathrm{SP}(1, 6)$: $z(\mathrm{SP}(1, 6)) = 13$, con

$$x_{13} = x_{32} = x_{24} = x_{46} = 1$$

  e tutte le altre variabili a 0: il cammino $1, 3, 2, 4, 6$.

- Il valore duale è $\pi_1 - \pi_6 = 13$.

Esempio (2/3)

- $\mathrm{SP}(1)$: $z(\mathrm{SP}(1)) = 0 + 5 + 2 + 8 + 10 + 13 = 38$, con i flussi sull'albero dei cammini minimi

$$x_{13} = 5, \qquad x_{32} = 3, \qquad x_{24} = 2, \qquad x_{46} = 1, \qquad x_{35} = 1.$$

- Per esempio $T_3 = \{2, 3, 4, 5, 6\}$, quindi $x_{13} = 5$; il costo è

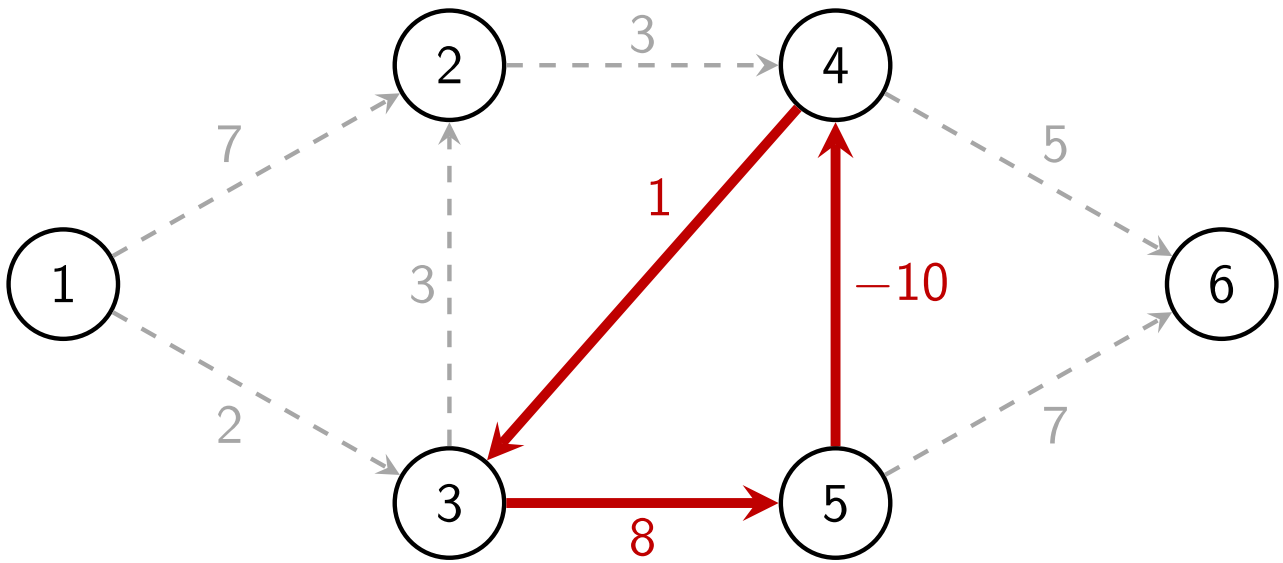
$$2 \cdot 5 + 3 \cdot 3 + 3 \cdot 2 + 5 + 8 = 38.$$

- I duali ottimi, normalizzati con $\pi_1 = 0$, sono $\boldsymbol{\pi} = (0, -5, -2, -8, -10, -13)$.

Esempio (3/3)

- Con le lunghezze d'' dell'esempio del ciclo negativo ($d''_{54} = -10$) il modello $\mathrm{SP}(1, 6)$ è illimitato.

- Ogni unità di flusso lungo il ciclo $3, 5, 4, 3$ abbassa il valore di 1.



Riepilogo

Il modello primale:

- Un cammino minimo è un flusso unitario da s a t: $\min \boldsymbol{d}'\boldsymbol{x}$ con $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq 0$
- Un ciclo negativo in un punto qualsiasi di G rende il modello illimitato

Il modello duale:

- I suoi vincoli sono $\pi_i - \pi_j \leq d_{ij}$: i potenziali ammissibili
- $\pi_i = -\tilde{d}(i)$ è ottimo, anche per il modello a sorgente singola $\mathrm{SP}(s)$

Cammini minimi nei grafi aciclici

Introduzione: Cammini minimi nei grafi aciclici

- Cominciamo gli algoritmi dal caso più facile: un grafo senza cicli orientati
- Ordinando i nodi in modo che ogni arco vada in avanti, una sola scansione degli archi dà tutte le distanze, con lunghezze di qualsiasi segno
- Lo stesso algoritmo calcola i cammini massimi, il cuore della gestione dei progetti più avanti nel corso

Definizione 4: grafo aciclico e ordine topologico

Un grafo orientato senza cicli orientati è un ***grafo aciclico orientato*** *(DAG). Un* ***ordine topologico*** *di $G = (N, A)$ è un ordinamento $v_1, v_2, \ldots, v_n$ dei suoi nodi tale che ogni arco va in avanti:*

$$(v_a, v_b) \in A \quad \Longrightarrow \quad a < b.$$

Proposizione 4: ordini topologici

Un grafo orientato ha un ordine topologico se e solo se è aciclico.

Dimostrazione

Se G ha un ordine topologico, un ciclo $i_0, i_1, \ldots, i_k = i_0$ andrebbe in avanti su ogni arco e tornerebbe al suo primo nodo: impossibile. Viceversa, sia G aciclico; ragioniamo per induzione su n.

1. G ha un nodo senza archi entranti: altrimenti, camminando all'indietro lungo archi entranti a partire da un nodo qualsiasi, dopo n passi un nodo si ripete, e la passeggiata contiene un ciclo.

2. Mettiamo un nodo di questo tipo v_1 per primo. Il grafo senza v_1 è aciclico, e per induzione ha un ordine topologico $v_2, \ldots, v_n$; nessun arco entra in v_1, quindi $v_1, v_2, \ldots, v_n$ è un ordine topologico di G.

□

Algorithm: Cammini minimi in un grafo aciclico

Input : un grafo aciclico $G = (N, A)$ con lunghezze d_{ij} di qualsiasi segno, una sorgente $s \in N$

Output: le distanze $d(j) = \tilde{d}(j)$ e i predecessori $pred(j)$

1 **Passo 0 (ordine).** Trova un ordine topologico: togli ripetutamente un nodo senza archi entranti (quello di indice minimo, se ce n'è più di uno).

2 Poni $d(s) := 0$ e $d(j) := +\infty$ per $j \neq s$.

3 **Passo 1 (etichette).** Per ogni nodo i in ordine topologico con $d(i) < +\infty$, e ogni arco $(i, j) \in \delta^+(i)$:

4 se $d(i) + d_{ij} < d(j)$, poni $d(j) := d(i) + d_{ij}$ e $pred(j) := i$.

5 STOP.

Teorema 4: correttezza e tempo di esecuzione sui grafi aciclici

Su un grafo aciclico con lunghezze di qualsiasi segno, l'algoritmo «Cammini minimi in un grafo aciclico» termina con $d(j) = \tilde{d}(j)$ *per ogni* $j \in N$*, e richiede tempo* $O(n + m)$*.*

- *Idea della dimostrazione:* i nodi di un cammino minimo compaiono nell'ordine topologico, quindi ciascuno viene elaborato dopo il suo predecessore sul cammino.

Dimostrazione (1/2)

Un grafo aciclico non ha alcun ciclo, quindi vale (A2). Ogni etichetta finita è la lunghezza di una passeggiata da s, quindi $d(j) \geq \tilde{d}(j)$ per il lemma su passeggiate e cammini; le etichette possono solo diminuire.

1. Sia j raggiungibile e $s = i_0, i_1, \ldots, i_k = j$ un cammino minimo; i suoi nodi compaiono in quest'ordine nell'ordine topologico, perché ogni arco va in avanti.

2. Per induzione su t, dopo l'elaborazione di i_t vale $d(i_t) \leq \tilde{d}(i_t)$: il cammino fino a i_{t-1} è minimo (ottimalità dei sottocammini), e l'elaborazione di i_{t-1} dà

$$d(i_t) \leq \tilde{d}(i_{t-1}) + d_{i_{t-1} i_t} = \tilde{d}(i_t).$$

Dimostrazione (2/2)

3. I nodi non raggiungibili da s conservano $d(j) = +\infty = \tilde{d}(j)$.

Tempo.

- L'ordine topologico richiede tempo $O(n + m)$ con un contatore degli archi entranti di ogni nodo, decrementato quando si toglie una coda.
- Il Passo 1 esamina ogni arco una volta.

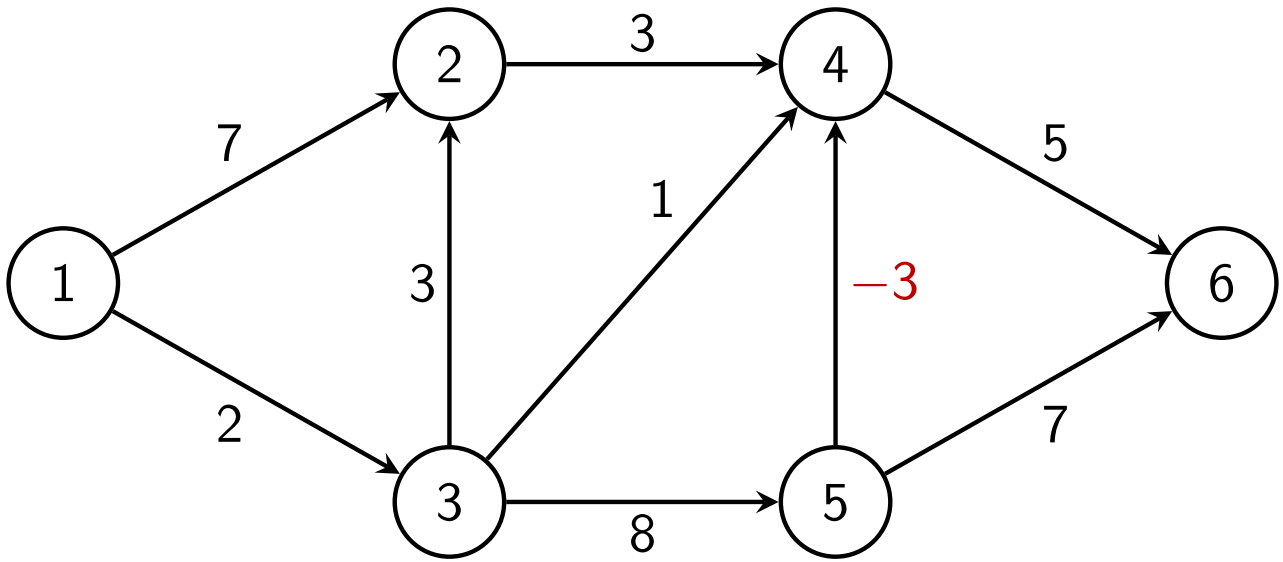
□

Osservazioni

- La regola dell'indice minimo fissa un ordine per gli esempi; si può togliere qualsiasi nodo senza archi entranti.
- Con le lunghezze $-d_{ij}$ l'algoritmo calcola i **cammini massimi**, perché un grafo aciclico non ha cicli da ripetere.
- I cammini massimi nei grafi aciclici sono il cuore del capitolo «Gestione dei progetti».

Esempio: un'istanza aciclica

Invertendo l'arco $(4, 3)$ del grafo di riferimento in $(3, 4)$, con $d_{34} = 1$, si ottiene un grafo aciclico. Gli diamo le lunghezze del primo esempio, tranne $d_{54} = -3$ (in rosso).



Esempio: Passo 0

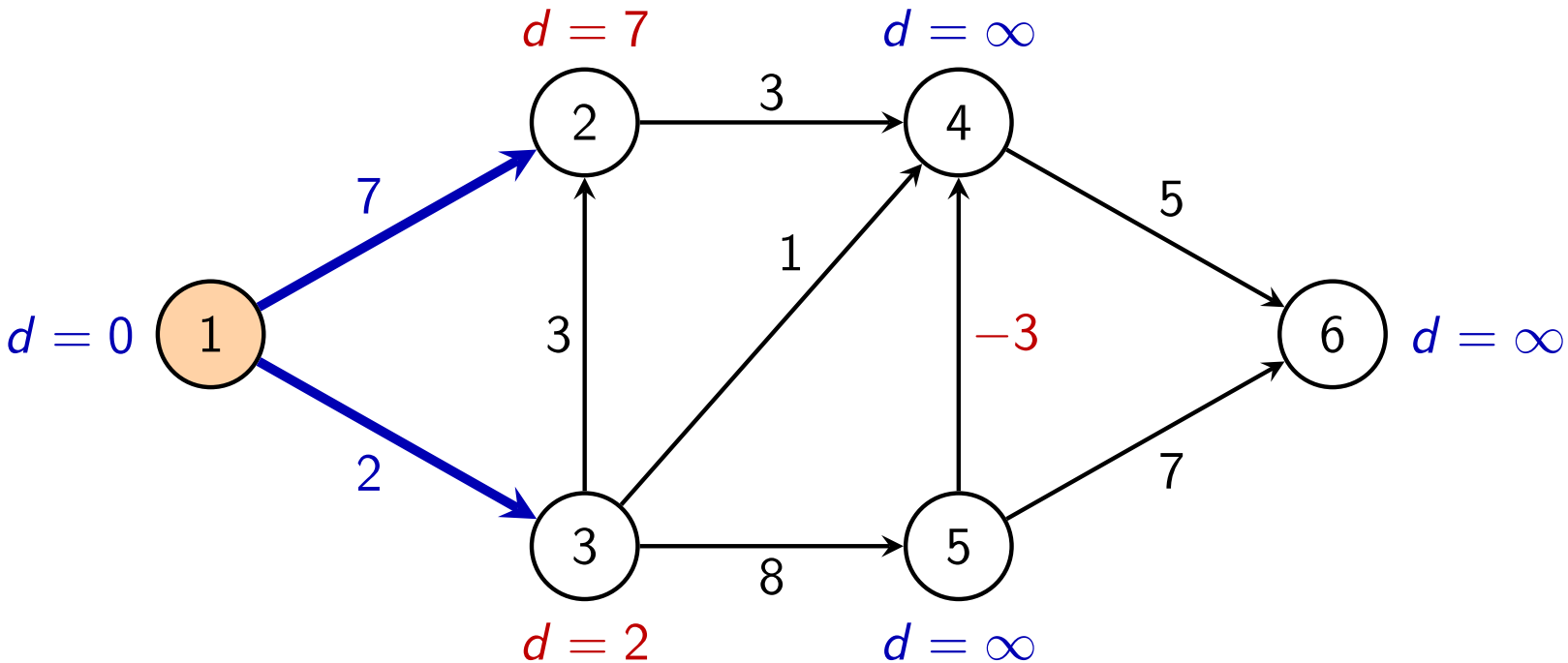
- Il Passo 0 dà l'ordine topologico

$$1,\ 3,\ 2,\ 5,\ 4,\ 6.$$

- Solo il nodo 1 non ha archi entranti; dopo 1, solo 3 non ha archi entranti.
- Poi 2 e 5 sono liberi, e 2 va per primo (indice minimo); poi 5, 4 e 6.
- $d(1) = 0$ e $d(j) = +\infty$ per $j \neq 1$.
- Nelle prossime slide il nodo che si elabora è in arancione, i nodi già elaborati sono ombreggiati, un'etichetta appena cambiata è in rosso, e gli archi $(pred(j), j)$ sono in blu.

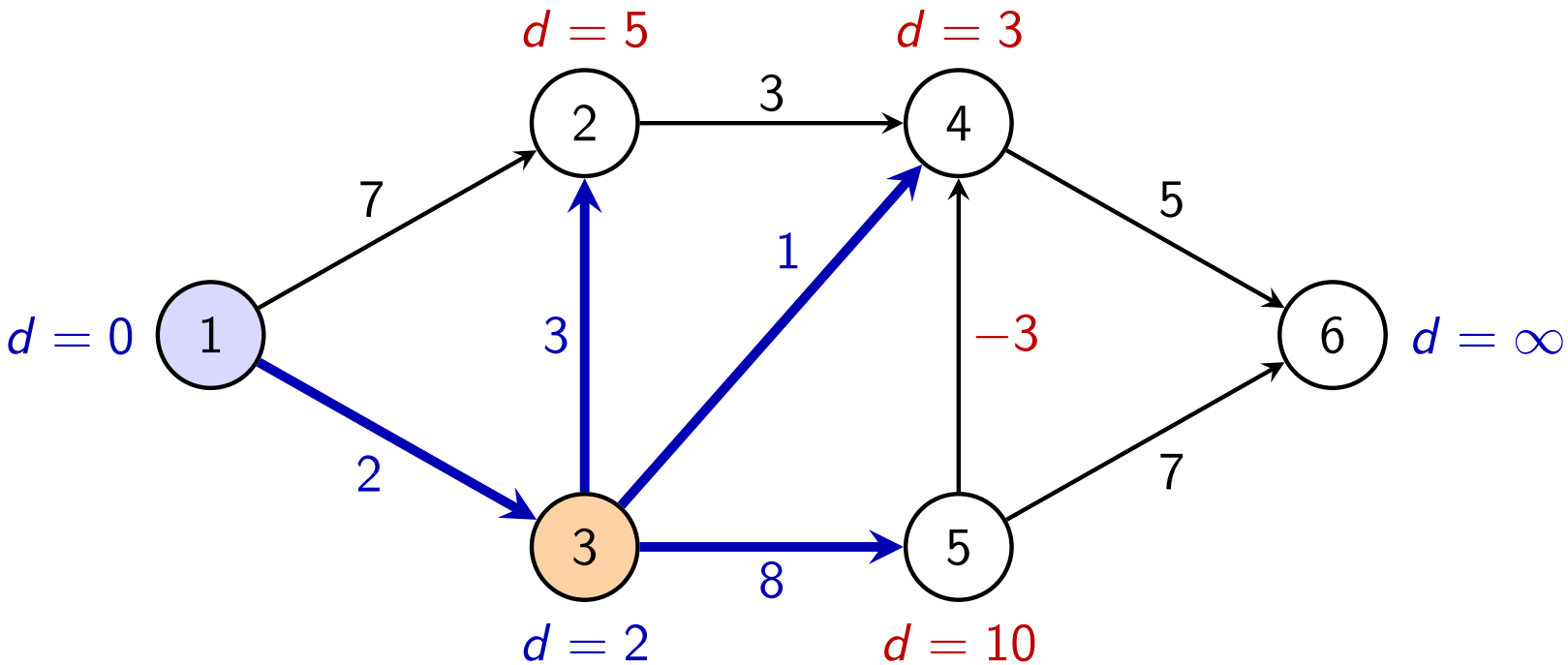
Esempio: passo 1, nodo 1

- $(1,2)$: $0+7=7<\infty$, quindi $d(2):=7$, $pred(2):=1$;
- $(1,3)$: $0+2=2<\infty$, quindi $d(3):=2$, $pred(3):=1$.



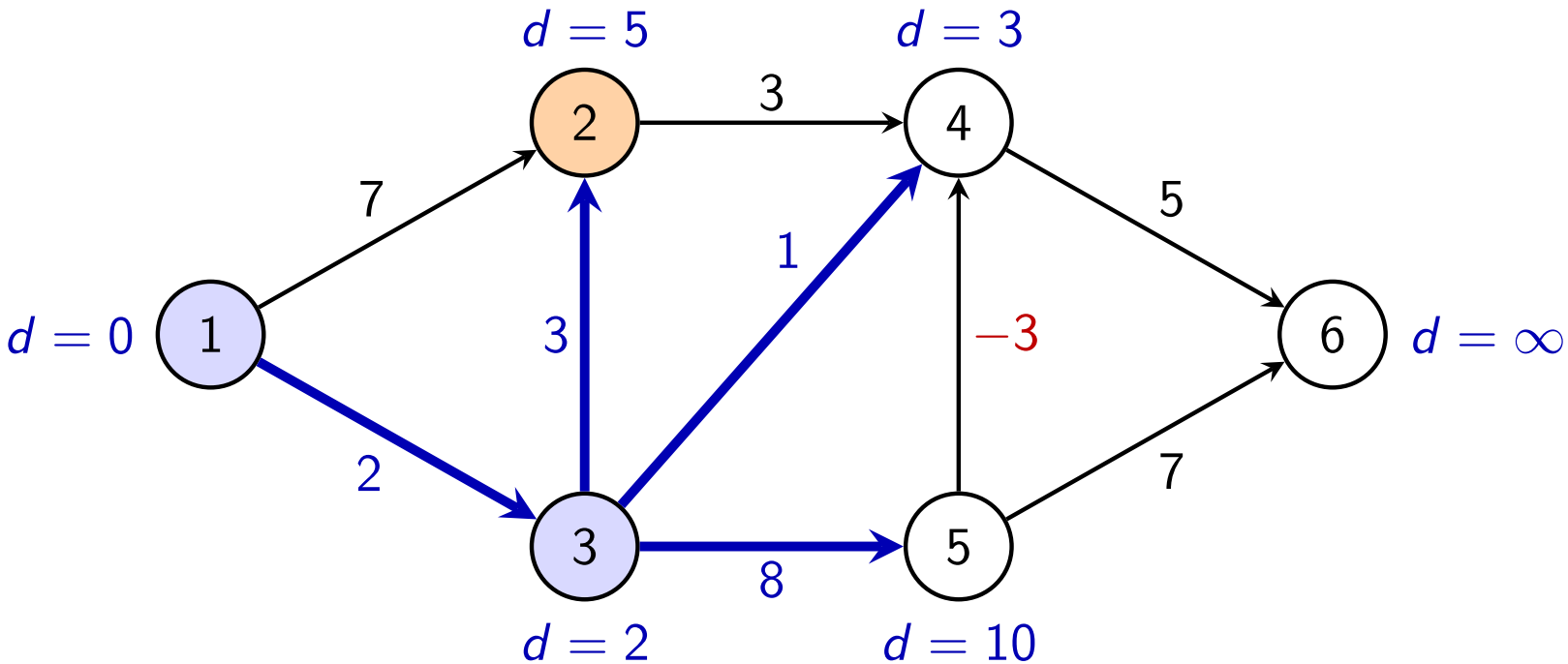
Esempio: passo 2, nodo 3

- $(3, 2)$: $2 + 3 = 5 < 7$, quindi $d(2) := 5$, $pred(2) := 3$;
- $(3, 4)$: $2 + 1 = 3 < \infty$, quindi $d(4) := 3$, $pred(4) := 3$;
- $(3, 5)$: $2 + 8 = 10 < \infty$, quindi $d(5) := 10$, $pred(5) := 3$.



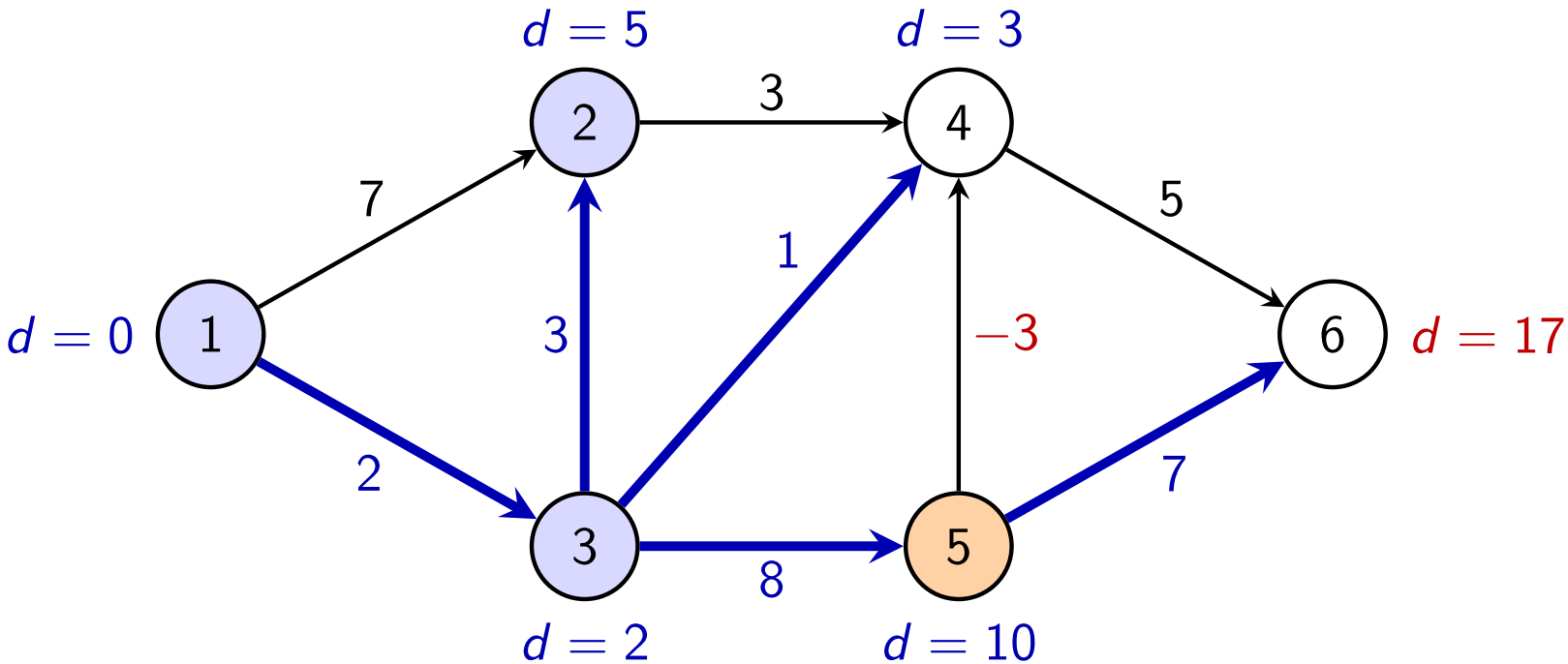
Esempio: passo 3, nodo 2

- $(2,4)$: $5+3=8 \geq 3=d(4)$: nessun cambiamento.



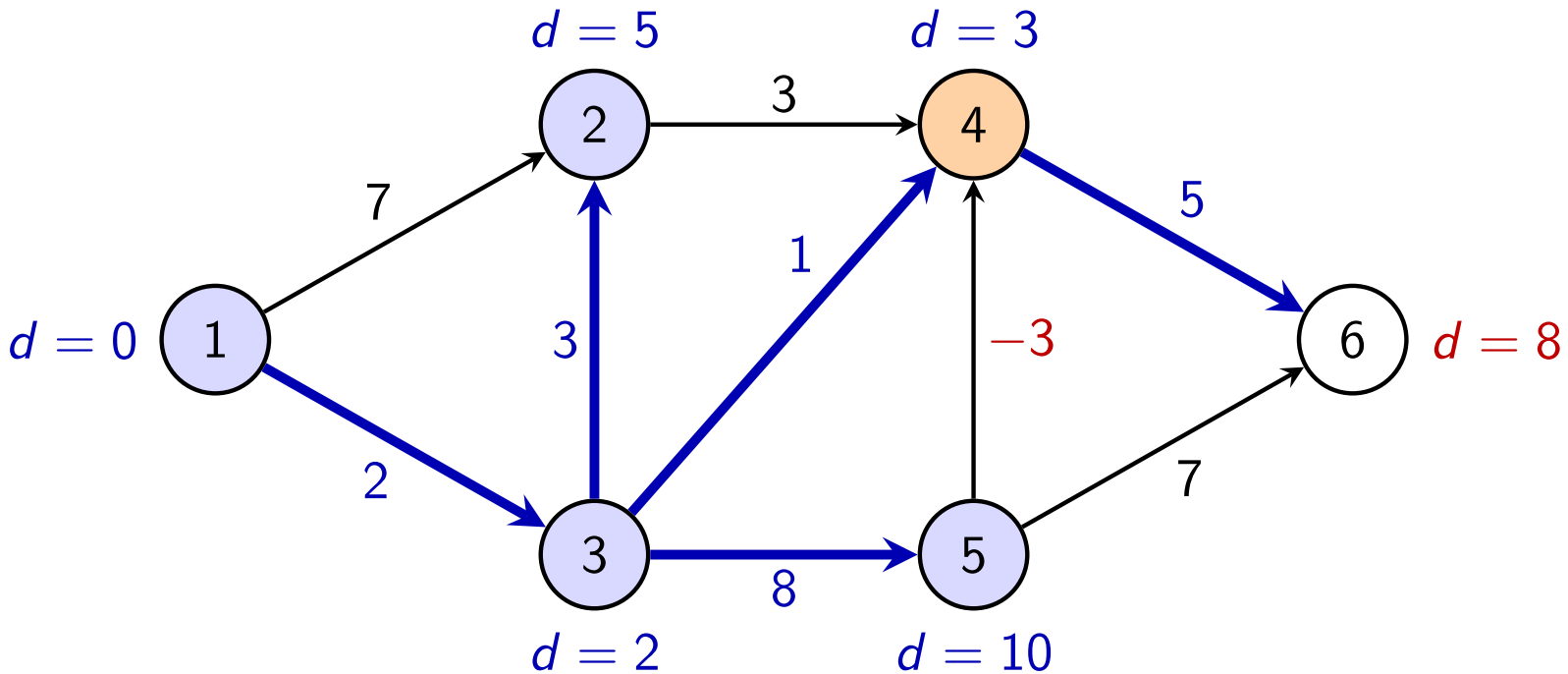
Esempio: passo 4, nodo 5

- $(5,4)$: $10-3=7\geq 3=d(4)$: nessun cambiamento;
- $(5,6)$: $10+7=17<\infty$, quindi $d(6):=17$, $pred(6):=5$.



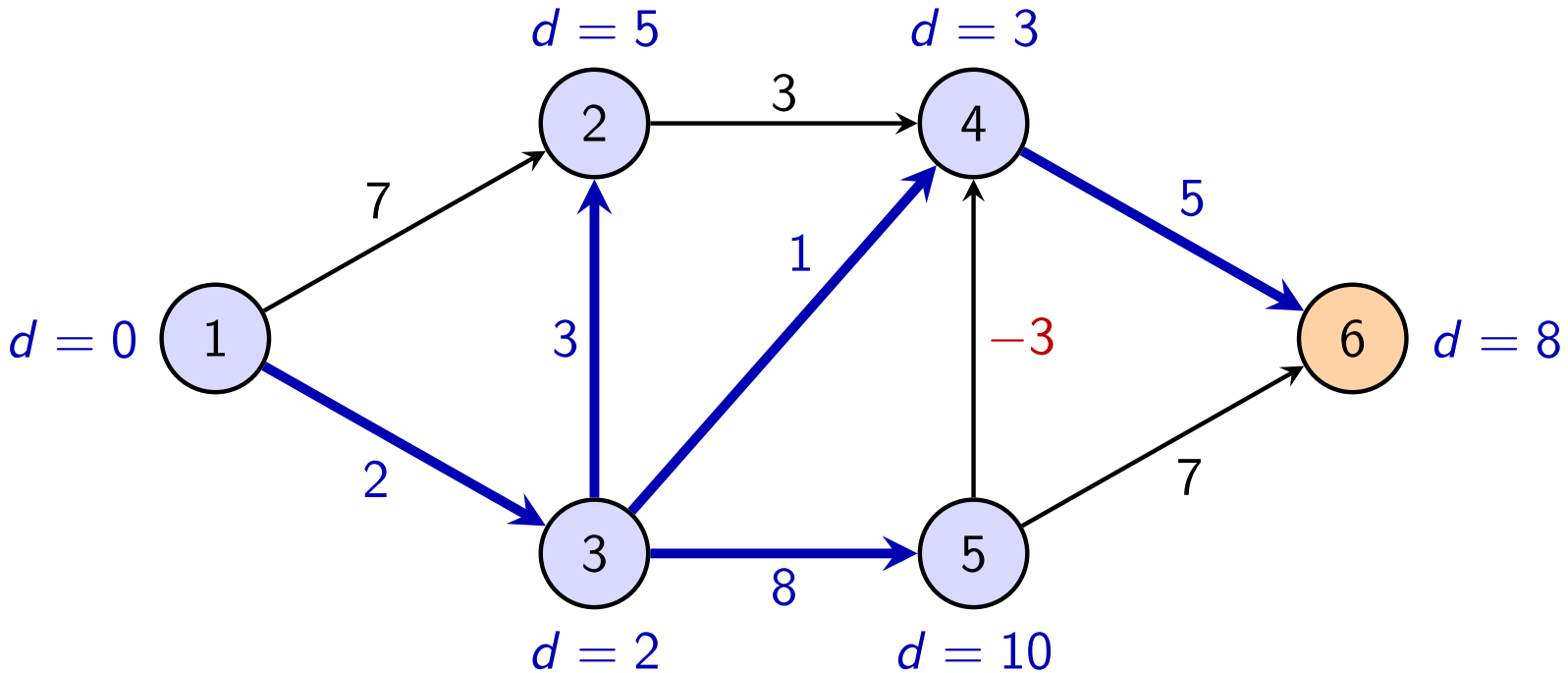
Esempio: passo 5, nodo 4

- $(4, 6)$: $3 + 5 = 8 < 17$, quindi $d(6) := 8$, $pred(6) := 4$.



Esempio: passo 6, nodo 6

- Il nodo 6 non ha archi uscenti: nulla cambia, e l'algoritmo si ferma.



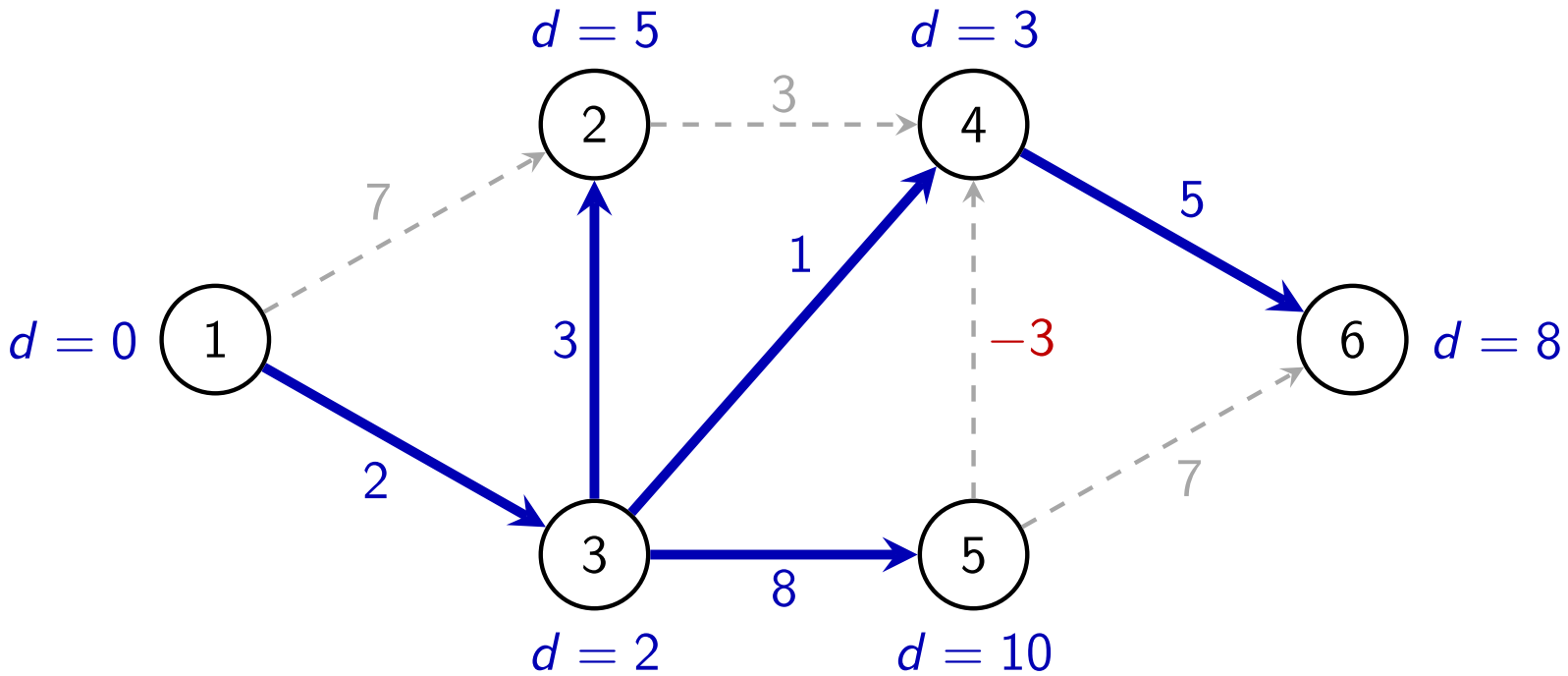
Esempio: la tabella delle etichette

Ogni riga della tabella elabora un nodo e mostra le etichette dopo l'elaborazione:

passo	nodo i	$d(1)$	$d(2)$	$d(3)$	$d(4)$	$d(5)$	$d(6)$
1	1	0	7	2	∞	∞	∞
2	3	0	5	2	3	10	∞
3	2	0	5	2	3	10	∞
4	5	0	5	2	3	10	17
5	4	0	5	2	3	10	8
6	6	0	5	2	3	10	8

Esempio: il risultato

I predecessori danno l'albero $(1,3)$, $(3,2)$, $(3,4)$, $(3,5)$, $(4,6)$ (in blu).



- L'arco negativo $(5,4)$ non viene usato: $d(5) - 3 = 7 > 3 = d(4)$.
- Con le lunghezze $-d_{ij}$ le lunghezze dei cammini massimi da 1 sono $(0, 7, 2, 10, 10, 17)$.

Riepilogo

Ordine topologico:

- Un grafo è aciclico se e solo se ha un ordine topologico
- Si trova togliendo ripetutamente un nodo senza archi entranti

L'algoritmo:

- Esaminando i nodi in ordine topologico, ogni etichetta è definitiva quando il suo nodo viene elaborato
- Cammini minimi in tempo $O(n+m)$, con lunghezze di qualsiasi segno; con $-d_{ij}$, cammini massimi

L'algoritmo di Bellman–Ford

Introduzione: L'algoritmo di Bellman–Ford

- Ora il grafo può avere cicli e le lunghezze possono essere negative
- L'algoritmo di Bellman–Ford calcola le passeggiate minime con al più $1, 2, \ldots$ archi, una passata alla volta
- Trova le distanze, oppure rileva un ciclo negativo raggiungibile dalla sorgente: è lo strumento generale quando qualche lunghezza è negativa

Definizione 5: etichette con al più k archi

Per $k \geq 0$ e $j \in N$, $d^k(j)$ è la lunghezza minima di una passeggiata da s a j con al più k archi, e $d^k(j) = +\infty$ se non ce n'è nessuna.

Lemma 3: ricorsione delle etichette

$d^0(s) = 0$, $d^0(j) = +\infty$ *per* $j \neq s$, *e per ogni* $k \geq 1$

$$d^k(j) = \min\left\{ d^{k-1}(j), \min_{(i,j)\in\delta^-(j)} \left(d^{k-1}(i) + d_{ij}\right)\right\}, \quad \forall j \in N.$$

Dimostrazione

- L'unica passeggiata con 0 archi è la passeggiata s.
- Per $k \geq 1$, una passeggiata verso j con al più k archi
  - ha al più $k-1$ archi, con lunghezza minima $d^{k-1}(j)$, oppure
  - ha esattamente k archi e ultimo arco (i,j): è una passeggiata verso i con $k-1$ archi seguita da (i,j), e la sua lunghezza minima è

$$\min_{(i,j)} \left(d^{k-1}(i) + d_{ij} \right).$$

□

Algorithm: Algoritmo di Bellman–Ford

Input : un grafo orientato $G = (N, A)$ con lunghezze d_{ij} di qualsiasi segno, una sorgente $s \in N$

Output: le distanze $\tilde{d}(j)$ e i predecessori, oppure un ciclo negativo raggiungibile da s

1 **Passo 0 (inizializzazione).** Poni $d^0(s) := 0$, $d^0(j) := +\infty$ per $j \neq s$, e $k := 1$.

2 **Passo 1 (passata k).** Calcola $d^k(j)$ per ogni $j \in N$ con la ricorsione delle etichette;

3 se $d^k(j) < d^{k-1}(j)$, poni $pred(j)$ uguale alla coda i dell'arco che realizza il minimo.

4 **Passo 2 (test).** Se $d^k = d^{k-1}$, `STOP`: $\tilde{d} = d^k$.

5 Se $k = n$, `STOP`: un ciclo negativo è raggiungibile da s. Altrimenti poni $k := k + 1$ e torna al Passo 1.

Come funzionano le passate

- Ogni passata dipende solo dalle etichette della passata precedente: se $d^k = d^{k-1}$, tutte le passate successive ripetono le stesse etichette, e l'algoritmo può fermarsi.

- Negli esempi, i pareggi nel minimo si risolvono con la coda i di indice minimo, e il predecessore cambia solo quando l'etichetta diminuisce strettamente.

Teorema 5: correttezza di Bellman–Ford

Se nessun ciclo negativo è raggiungibile da s, allora $d^{n-1}(j) = \tilde{d}(j)$ per ogni $j \in N$, e l'algoritmo si ferma a una passata $k \leq n$ con $d^k = d^{k-1} = \tilde{d}$.

Dimostrazione

- Un cammino ha al più $n-1$ archi, quindi $d^{n-1}(j) \leq \tilde{d}(j)$.
- Per il lemma su passeggiate e cammini (che richiede solo i cicli raggiungibili da s) ogni passeggiata da s a j contiene un cammino non più lungo, quindi $d^k(j) \geq \tilde{d}(j)$ per ogni k.
- Quindi $d^{n-1} = d^n = \tilde{d}$.
- Le etichette non aumentano mai, e l'algoritmo si ferma alla prima passata k con $d^k = d^{k-1}$, che è al più n; da quella passata in poi le etichette non cambiano più, quindi $d^k = d^{n-1} = \tilde{d}$.

□

Teorema 6: rilevamento di un ciclo negativo

Un ciclo negativo è raggiungibile da s se e solo se $d^n \neq d^{n-1}$. In tal caso, sia $d^n(j) < d^{n-1}(j)$; leggendo i predecessori all'indietro a partire da j, una riga per passata, partendo dalla riga n della tabella, si ottiene una passeggiata da s a j con n archi che contiene un ciclo negativo.

Dimostrazione (1/2)

Se nessun ciclo negativo è raggiungibile da s, allora $d^n = d^{n-1}$ per la correttezza di Bellman–Ford. Viceversa, sia $d^n(j) < d^{n-1}(j)$.

1. Ogni passeggiata verso j di lunghezza $d^n(j)$ con al più n archi ha esattamente n archi, altrimenti $d^{n-1}(j) \leq d^n(j)$.

2. Sia i il predecessore di j nella riga n, quindi $d^n(j) = d^{n-1}(i) + d_{ij}$. Una passeggiata che realizza $d^{n-1}(i)$ seguita da (i, j) realizza $d^n(j)$, quindi ha esattamente $n - 1$ archi: $d^{n-1}(i) < d^{n-2}(i)$, e i ha un predecessore nella riga $n - 1$.

Dimostrazione (2/2)

3. Ripetendo il ragionamento fino alla riga 1, il cui predecessore è s, si ottiene una passeggiata W da s a j con n archi e $d(W) = d^n(j)$.

4. W ha $n+1$ nodi, quindi ripete un nodo e contiene un ciclo C. Togliendo C si ottiene una passeggiata verso j con meno di n archi, di lunghezza

$$d(W) - d(C) \geq d^{n-1}(j) > d^n(j) = d(W).$$

Quindi $d(C) < 0$, e C è raggiungibile da s lungo W.

□

Proposizione 5: l'albero dei predecessori di Bellman–Ford

Se nessun ciclo negativo è raggiungibile da s, alla fine gli archi $(pred(j), j)$, per i nodi $j \neq s$ con etichetta finita, formano un albero dei cammini minimi dei nodi raggiungibili.

Dimostrazione

Sia $d = \tilde{d}$ l'etichetta finale, e sia $t(j)$ l'ultima passata in cui l'etichetta di j è diminuita; $t(s) = 0$, perché $d^k(s) = 0$ per ogni k. Sia $p = pred(j)$.

1. Nella passata $t(j)$, $d(j) = d^{t(j)-1}(p) + d_{pj} \geq d(p) + d_{pj}$, perché le etichette non aumentano mai.

2. Per le equazioni di Bellman, $d(j) \leq d(p) + d_{pj}$. Quindi $d(j) = d(p) + d_{pj}$ e $d(p) = d^{t(j)-1}(p)$: l'etichetta di p non è cambiata dopo la passata $t(j) - 1$, quindi $t(p) < t(j)$.

3. Passando al predecessore t diminuisce strettamente, quindi la sequenza termina in s, lungo archi stretti: il cammino dell'albero da s a j ha lunghezza $d(j)$.

□

Teorema 7: tempo di esecuzione di Bellman–Ford

L'algoritmo di Bellman–Ford richiede tempo $O(n\,m)$.

Dimostrazione.

- Una passata calcola $d^k(j)$ per ogni j esaminando la stella entrante $\delta^-(j)$, quindi esamina ogni arco una volta: tempo $O(n+m)$.
- Le passate sono al più n, e il totale è $O(n(n+m))$, cioè $O(n\,m)$ quando $m \geq n-1$, come sotto (A1).

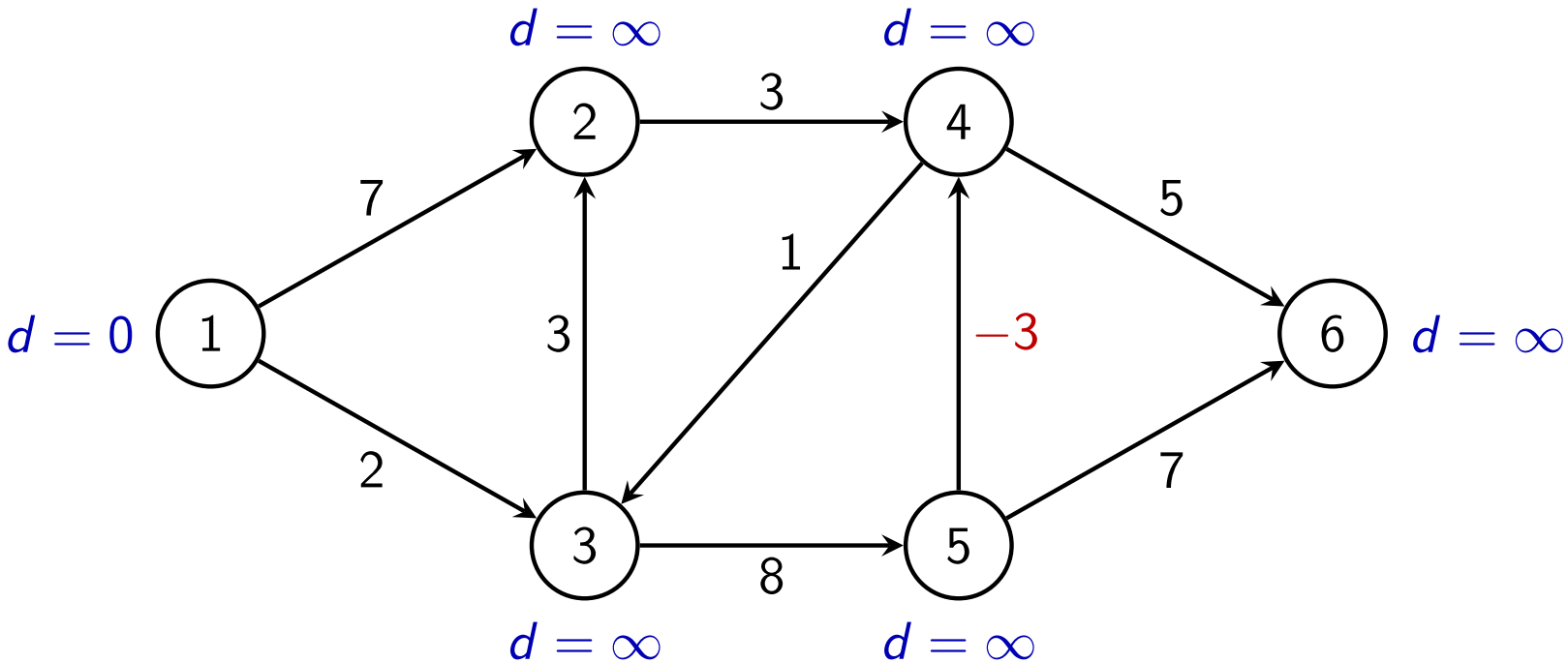
□

Esempio: Bellman–Ford con un arco negativo

- Usiamo le lunghezze d' del primo esempio con $d'_{54} = -3$ al posto di 2.
- I due cicli hanno lunghezze 7 e $8 - 3 + 1 = 6$, quindi non c'è alcun ciclo negativo.
- Nelle prossime slide ogni passata calcola le etichette d^k a partire dalle etichette d^{k-1} della passata precedente; un'etichetta appena diminuita è in rosso, e gli archi $(pred(j), j)$ sono in blu.

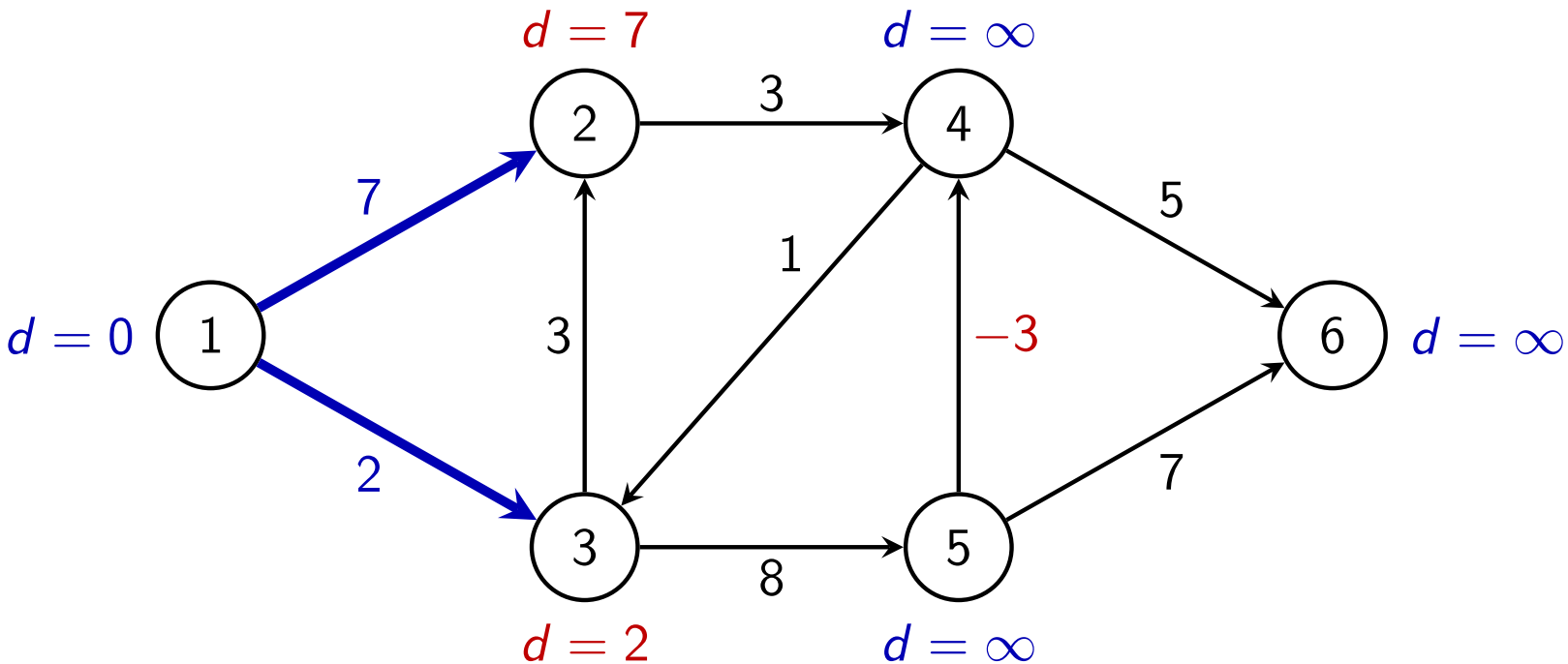
Esempio: passata 0

- $d^0 = (0, \infty, \infty, \infty, \infty, \infty)$.



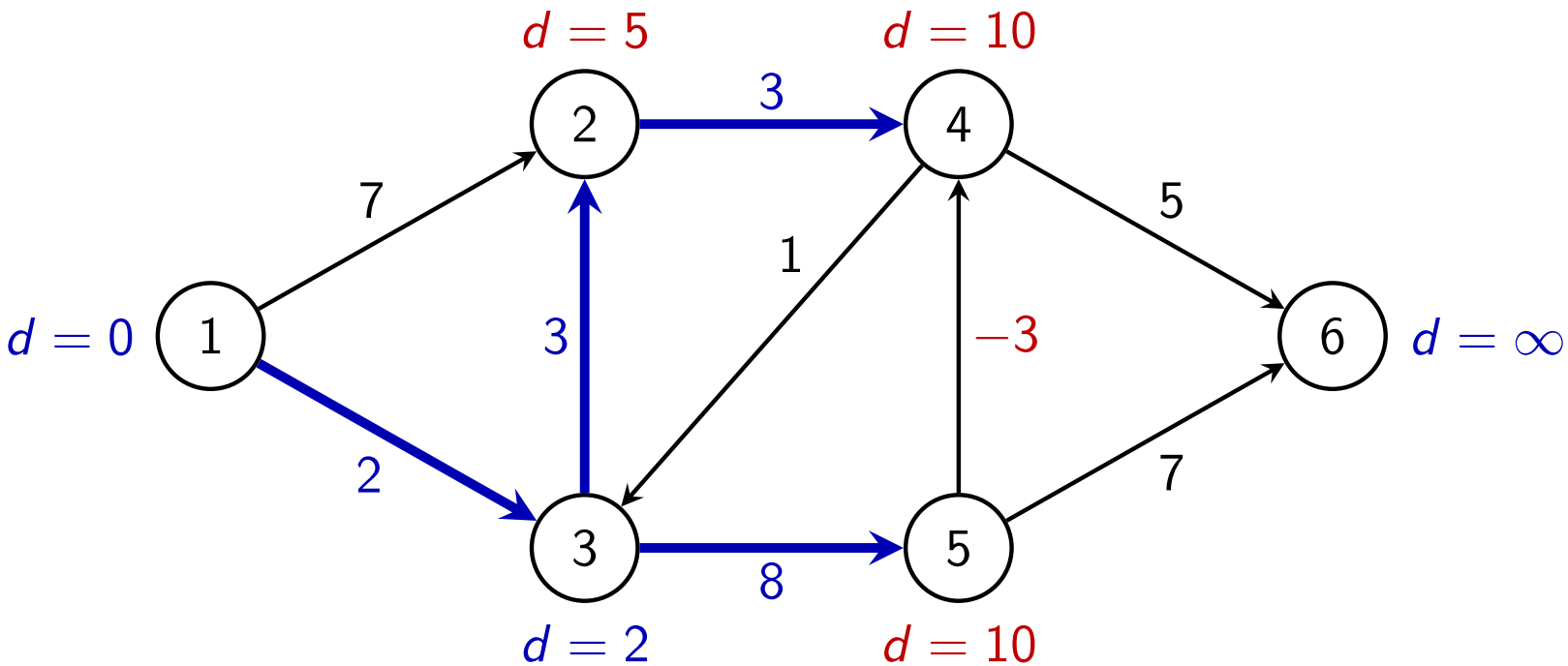
Esempio: passata 1

- $d^1(2) = \min\{\infty,\ 0+7\} = 7$, $pred(2) := 1$;
- $d^1(3) = \min\{\infty,\ 0+2\} = 2$, $pred(3) := 1$.



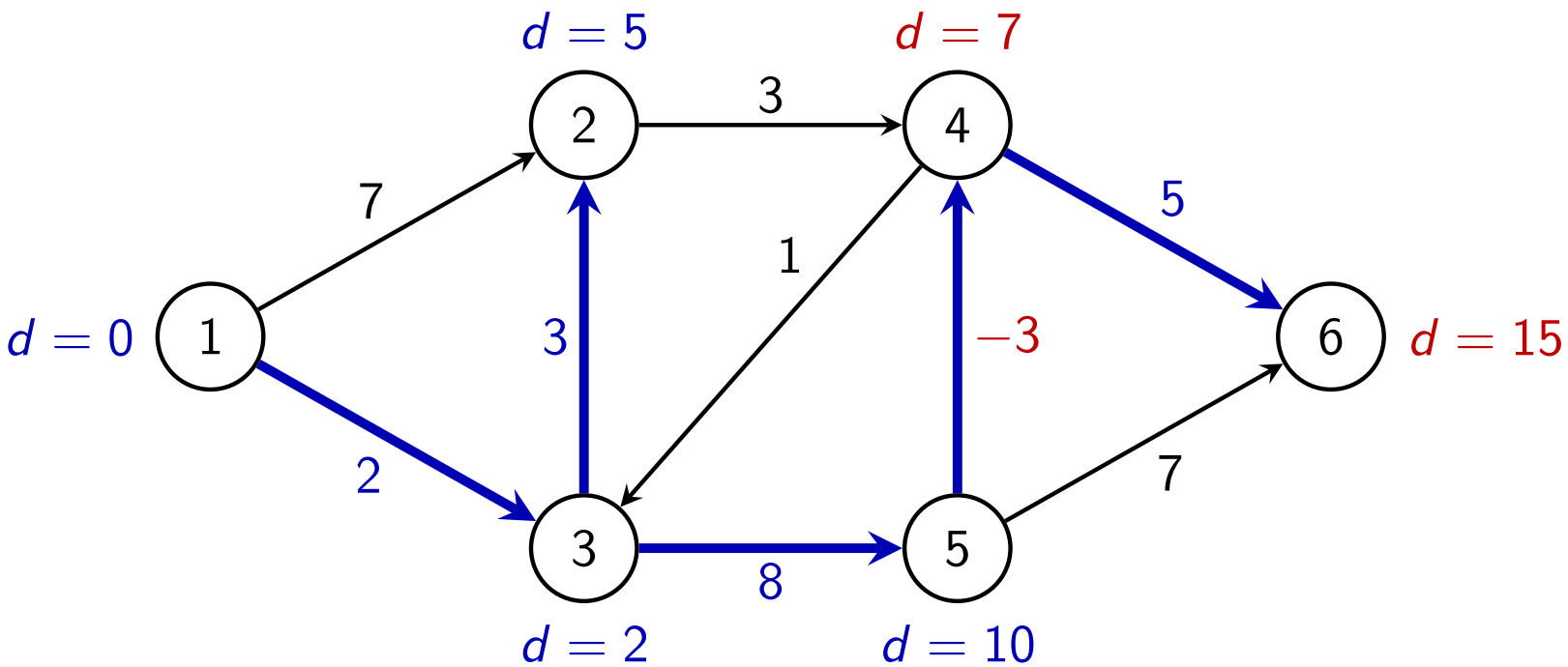
Esempio: passata 2

- $d^2(2) = \min\{7,\ 0+7,\ 2+3\} = 5$, $pred(2) := 3$;
- $d^2(4) = \min\{\infty,\ 7+3,\ \infty-3\} = 10$, $pred(4) := 2$;
- $d^2(5) = \min\{\infty,\ 2+8\} = 10$, $pred(5) := 3$.



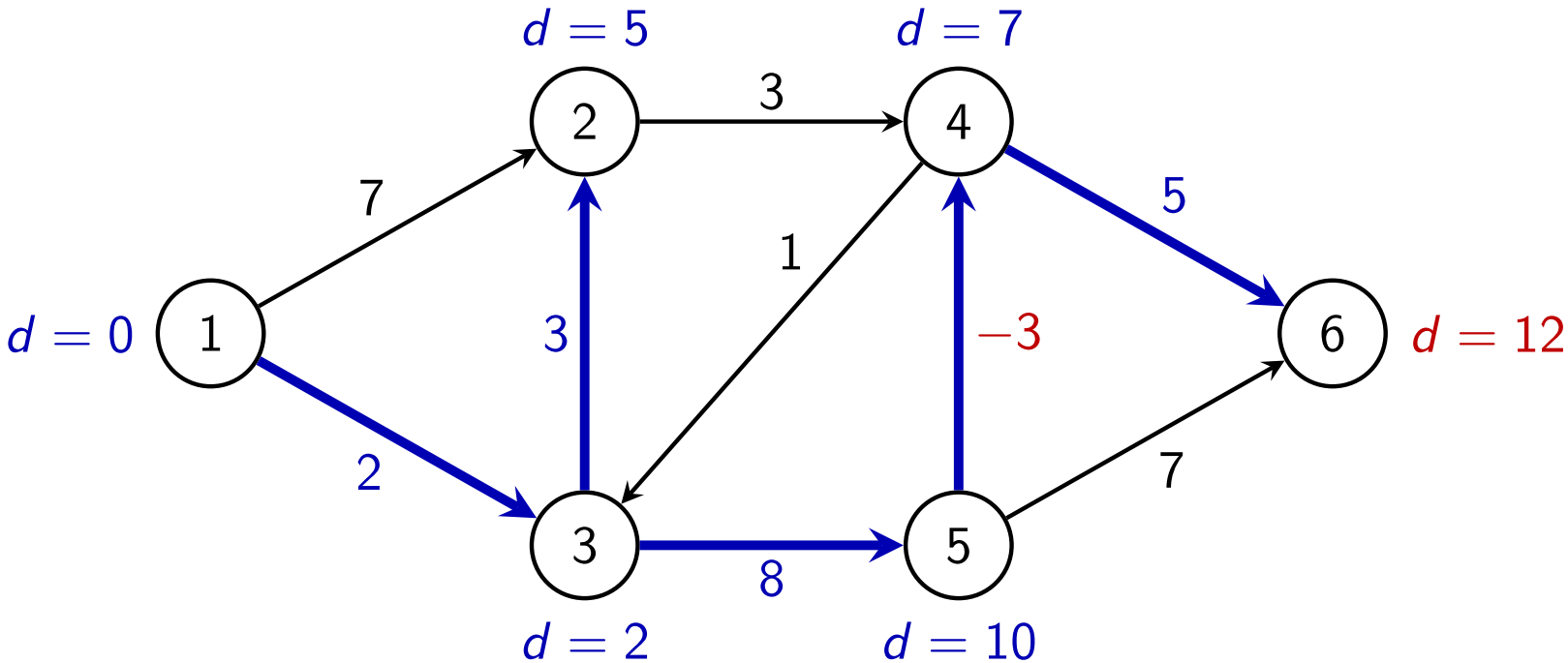
Esempio: passata 3

- $d^3(4) = \min\{10,\ d^2(2)+3,\ d^2(5)-3\} = \min\{10, 8, 7\} = 7$, con predecessore 5;
- $d^3(6) = \min\{\infty,\ 10+5,\ 10+7\} = 15$, $pred(6) := 4$.



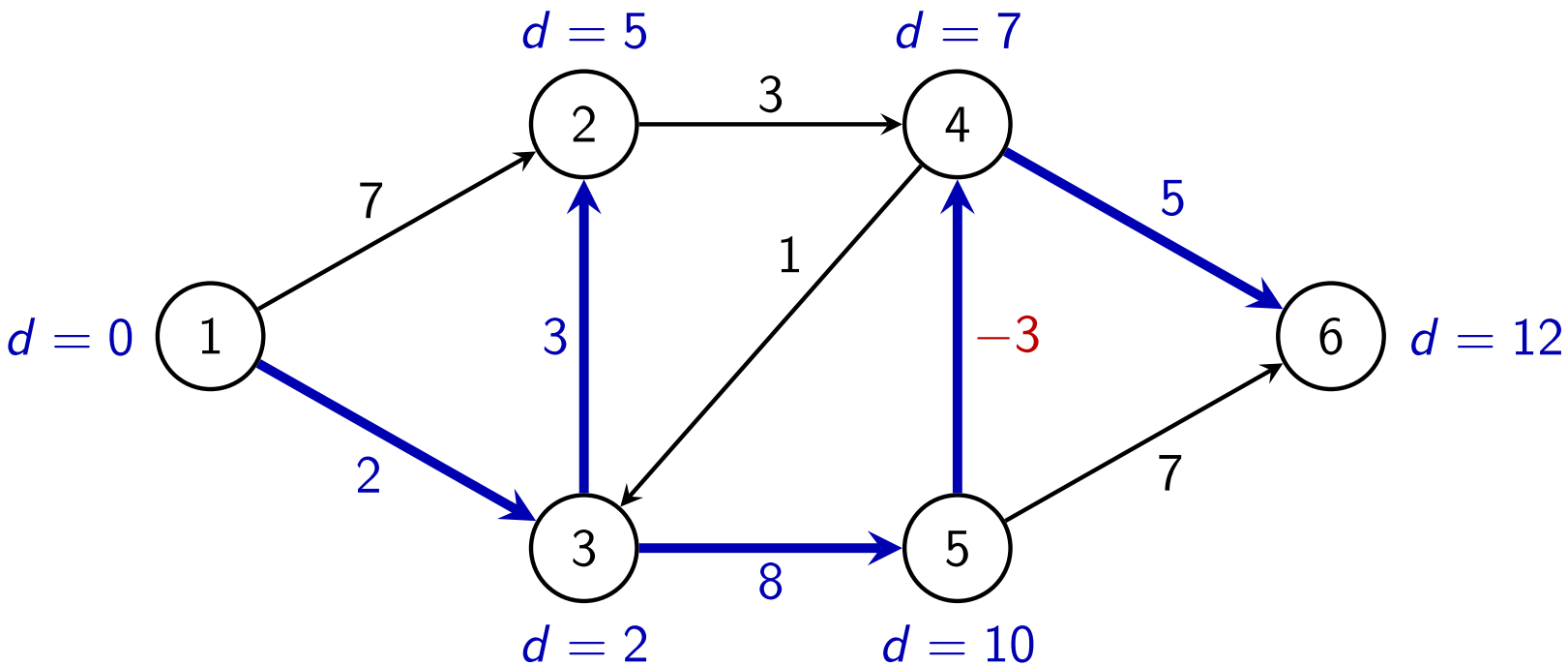
Esempio: passata 4

- $d^4(6) = \min\{15,\ 7+5,\ 10+7\} = 12$, $pred(6) = 4$ (invariato); nessun'altra etichetta diminuisce.



Esempio: passata 5

- Nessuna etichetta diminuisce: $d^5 = d^4$, e l'algoritmo si ferma con $\tilde{d} = (0, 5, 2, 7, 10, 12)$, dopo 5 passate invece delle $n = 6$ consentite.



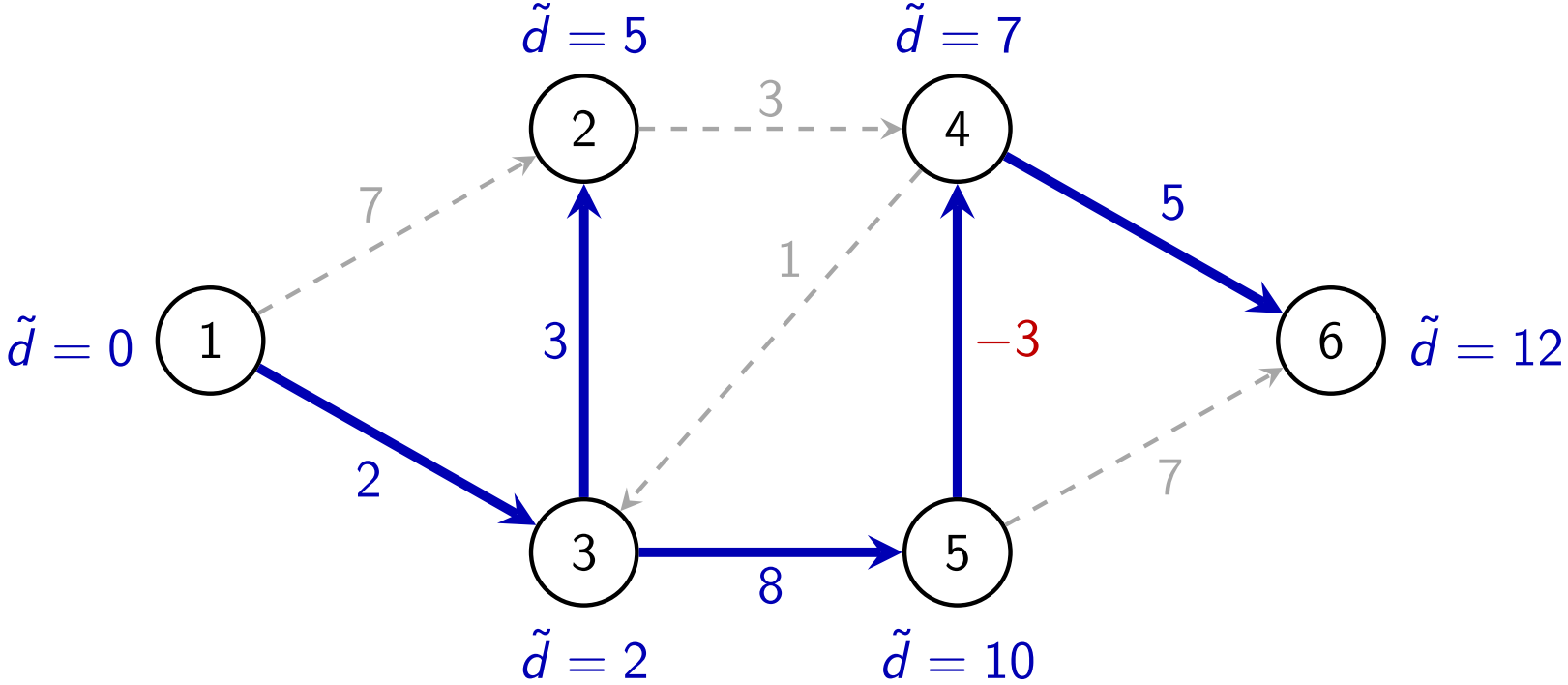
Esempio: la tabella delle passate

Ogni riga è una passata k; accanto a un'etichetta diminuita, tra parentesi il nuovo predecessore:

k	$d^k(1)$	$d^k(2)$	$d^k(3)$	$d^k(4)$	$d^k(5)$	$d^k(6)$
0	0	∞	∞	∞	∞	∞
1	0	7 (1)	2 (1)	∞	∞	∞
2	0	5 (3)	2	10 (2)	10 (3)	∞
3	0	5	2	7 (5)	10	15 (4)
4	0	5	2	7	10	12 (4)
5	0	5	2	7	10	12

Esempio: il risultato

L'albero dei cammini minimi ha gli archi $(1,3)$, $(3,2)$, $(3,5)$, $(5,4)$, $(4,6)$, e il cammino minimo verso 6 è $1,3,5,4,6$, di lunghezza $2+8-3+5=12$.

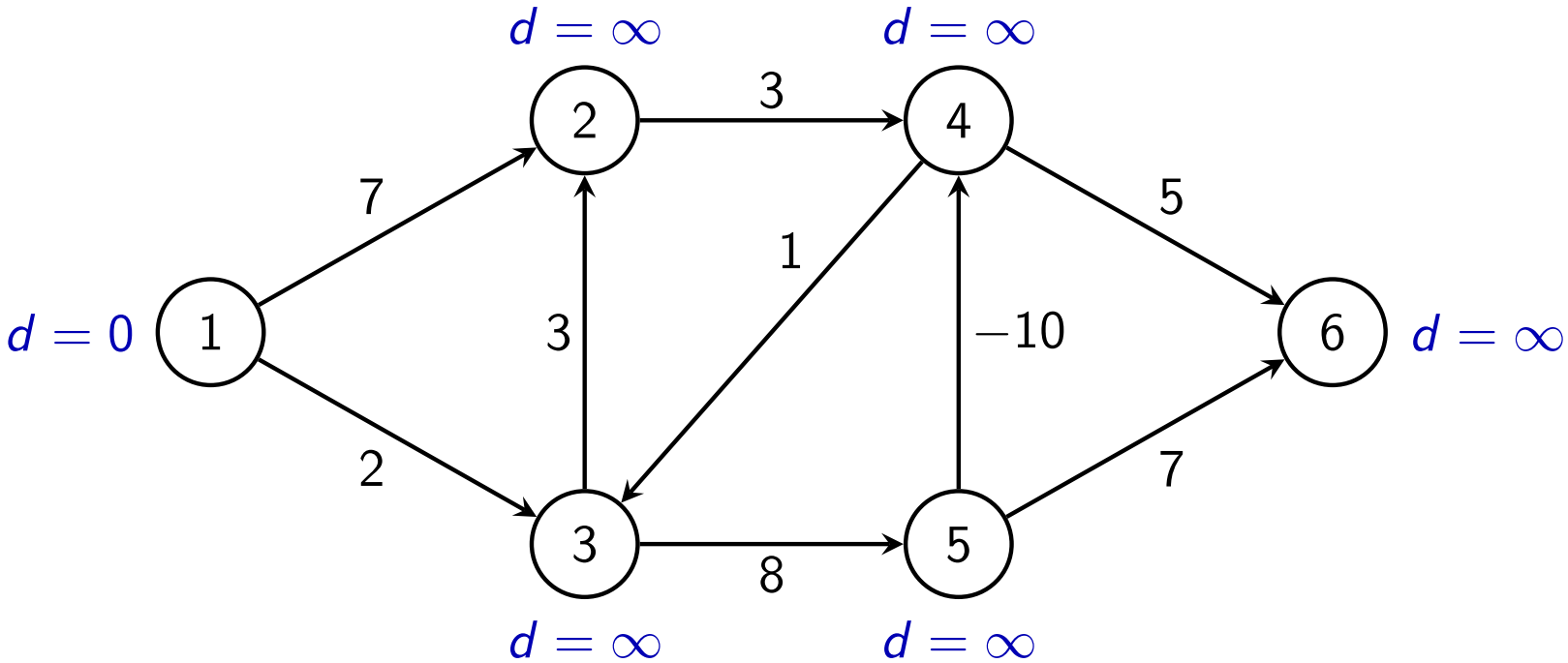


Che cosa succede quando un ciclo negativo è raggiungibile dalla sorgente?

Eseguiamo di nuovo l'algoritmo, con $d''_{54} = -10$.

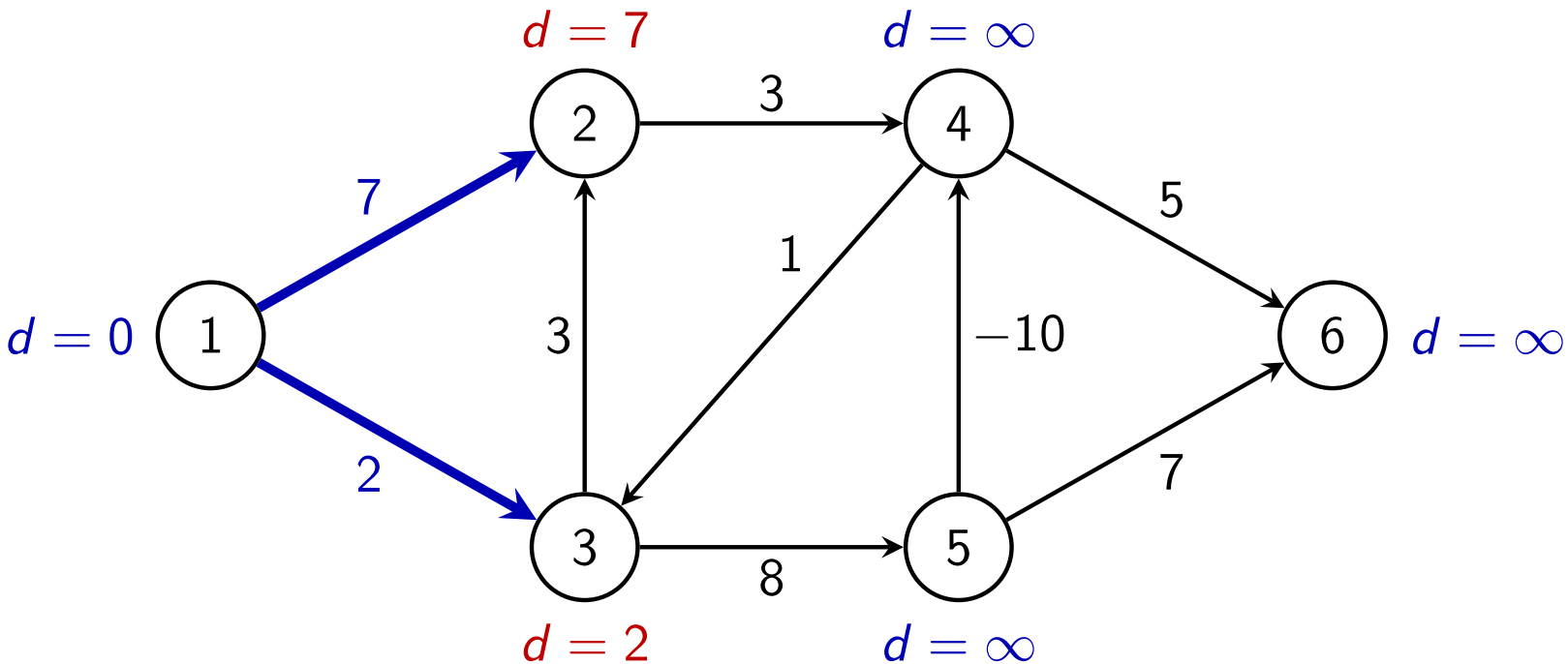
Esempio: Bellman–Ford rileva un ciclo negativo

- Con le lunghezze d'' dell'esempio del ciclo negativo ($d''_{54} = -10$) l'algoritmo esegue tutte le $n = 6$ passate.
- Passata 0: $d^0 = (0, \infty, \infty, \infty, \infty, \infty)$.



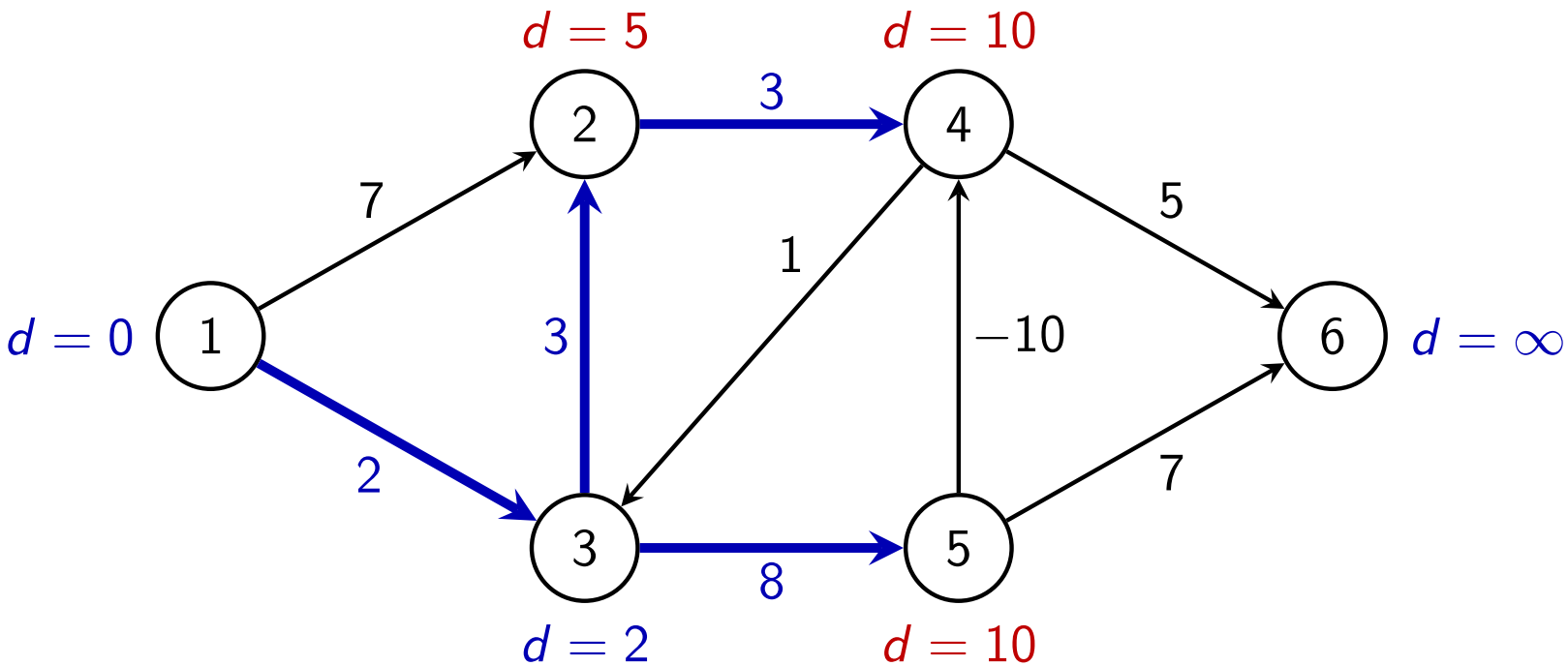
Esempio: passata 1

- $d^1(2) = \min\{\infty,\ 0+7\} = 7$, $pred(2) := 1$;
- $d^1(3) = \min\{\infty,\ 0+2\} = 2$, $pred(3) := 1$.



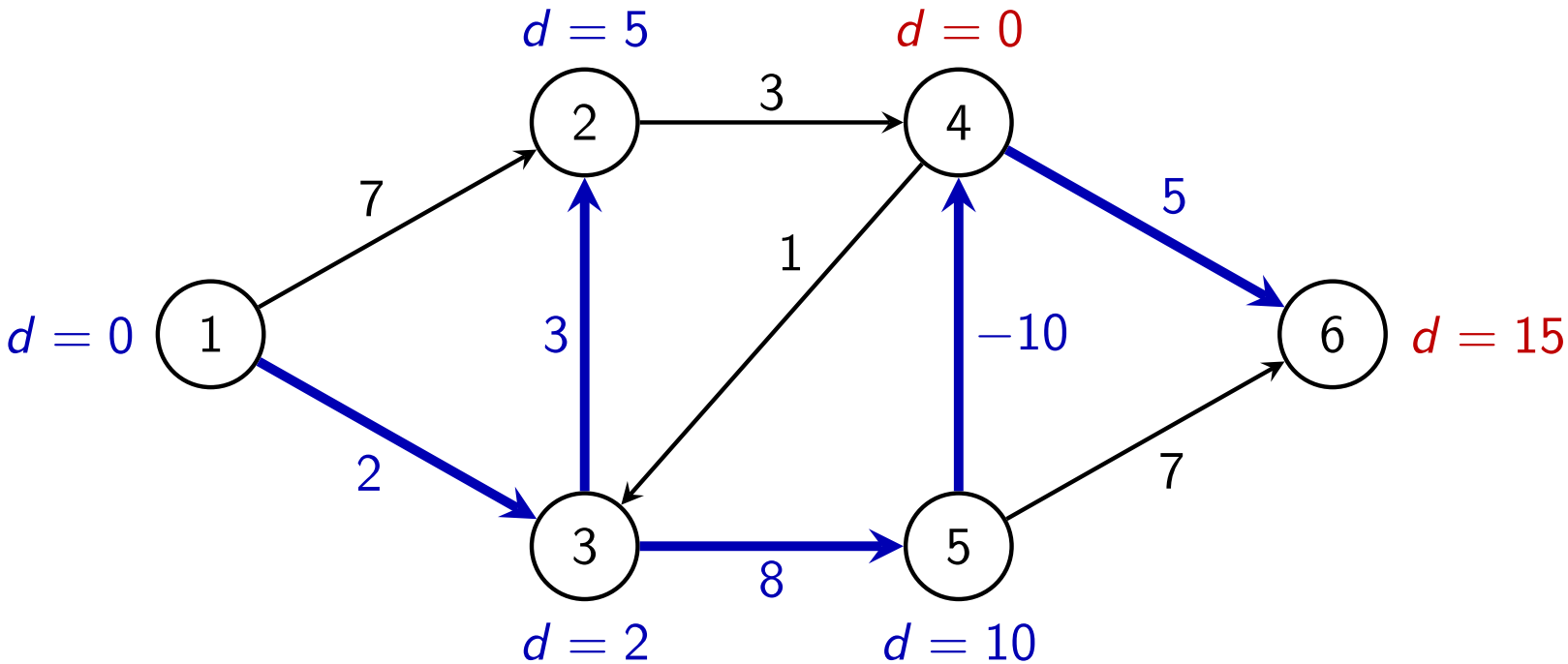
Esempio: passata 2

- $d^2(2) = \min\{7,\ 0+7,\ 2+3\} = 5$, $pred(2) := 3$;
- $d^2(4) = \min\{\infty,\ 7+3,\ \infty-10\} = 10$, $pred(4) := 2$;
- $d^2(5) = \min\{\infty,\ 2+8\} = 10$, $pred(5) := 3$.



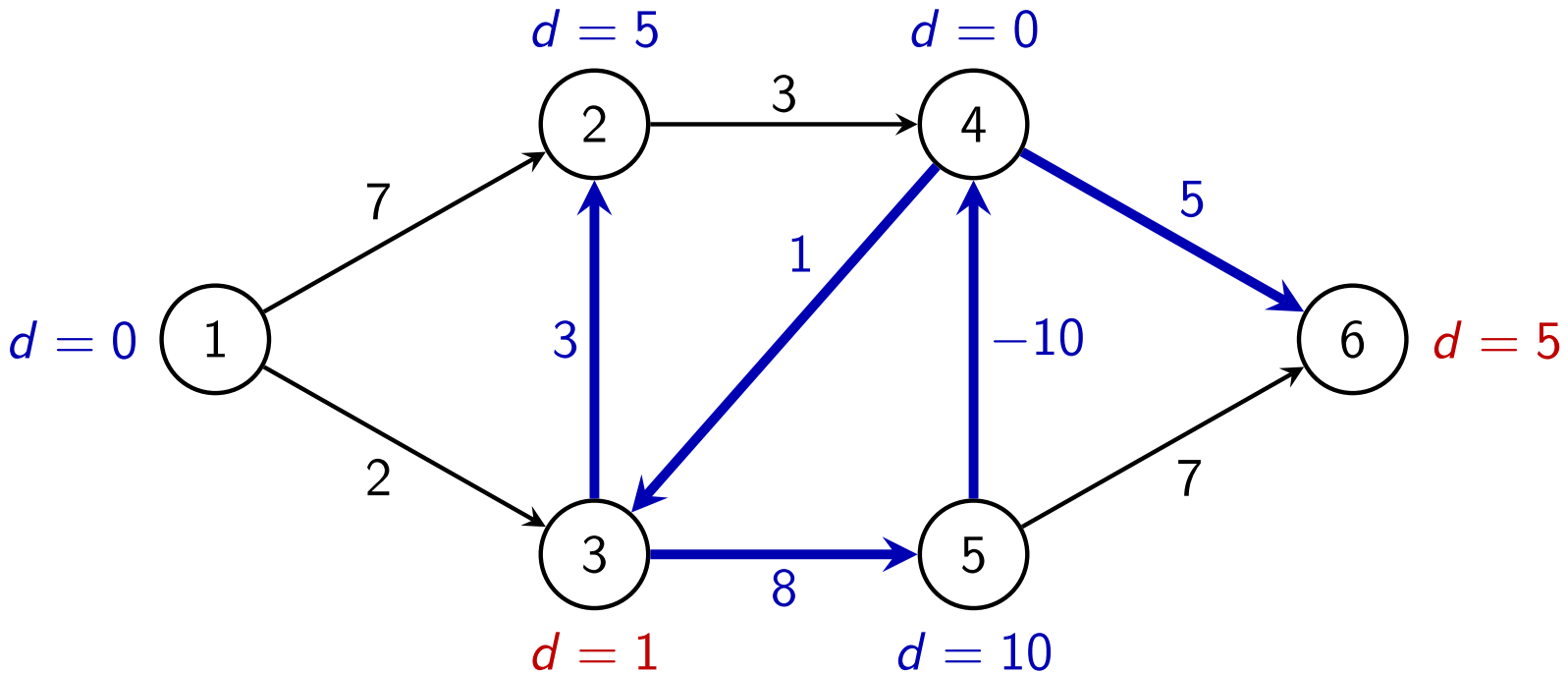
Esempio: passata 3

- $d^3(4) = \min\{10,\ 5+3,\ 10-10\} = 0$, $pred(4) := 5$;
- $d^3(6) = \min\{\infty,\ 10+5,\ 10+7\} = 15$, $pred(6) := 4$.



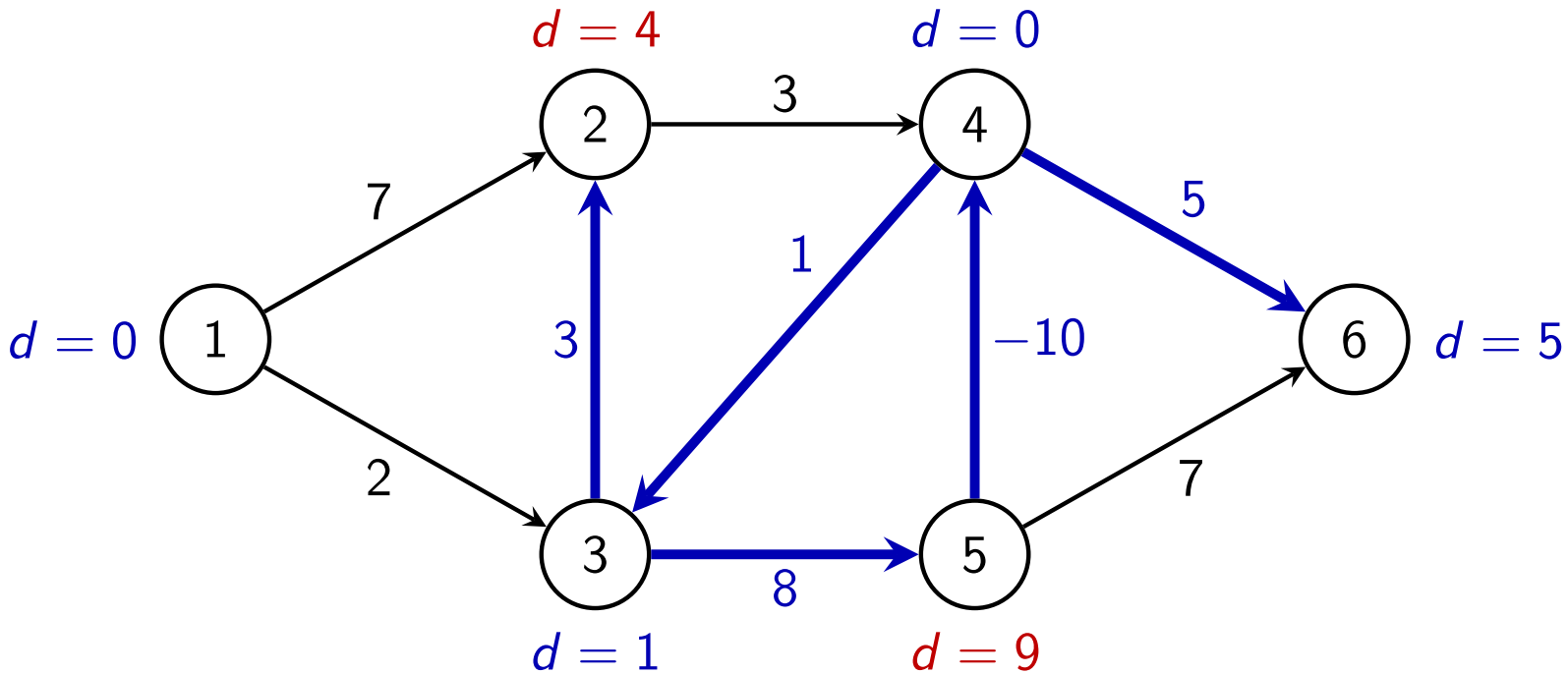
Esempio: passata 4

- $d^4(3) = \min\{2,\ 0+2,\ 0+1\} = 1$, $pred(3) := 4$;
- $d^4(6) = \min\{15,\ 0+5,\ 10+7\} = 5$, $pred(6) = 4$ (invariato).



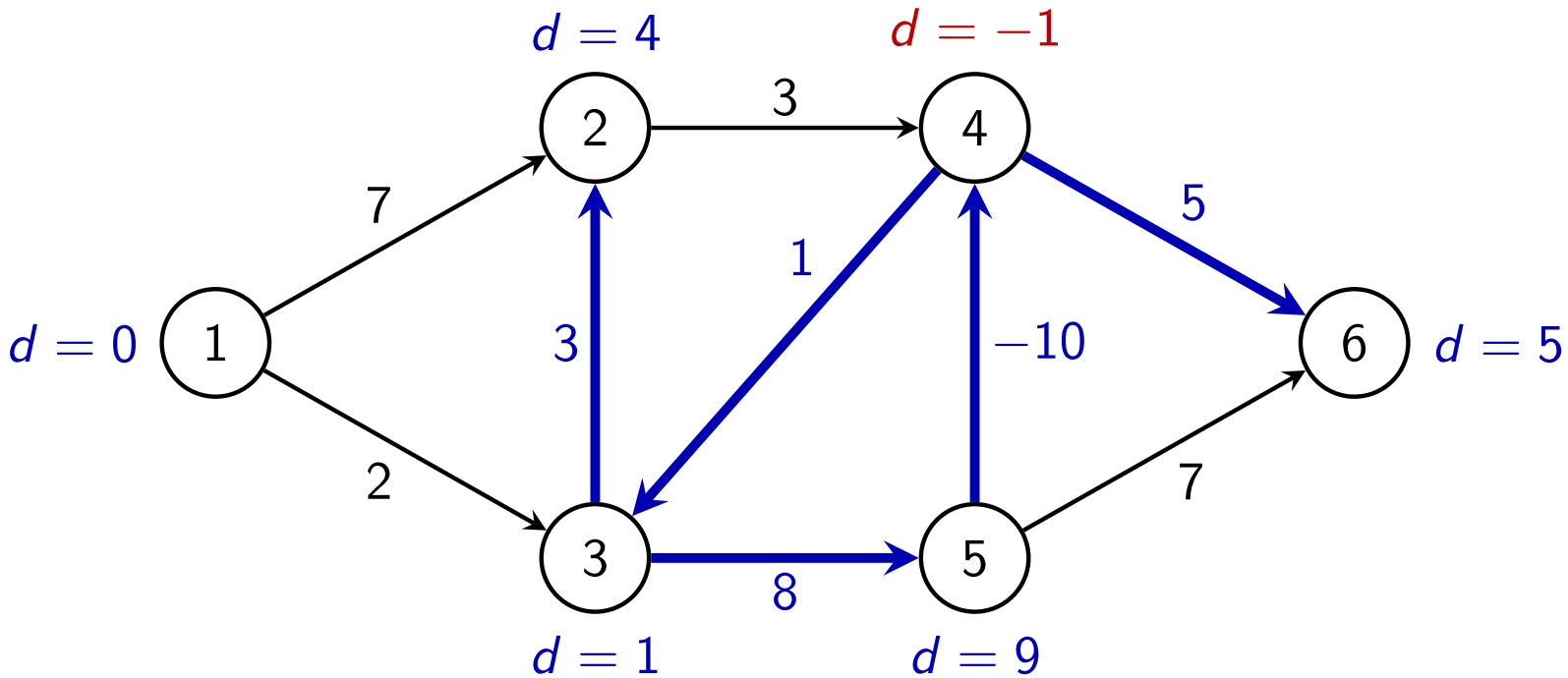
Esempio: passata 5

- $d^5(2) = \min\{5,\ 0+7,\ 1+3\} = 4$, $pred(2) = 3$ (invariato);
- $d^5(5) = \min\{10,\ 1+8\} = 9$, $pred(5) = 3$ (invariato).



Esempio: passata 6

- $d^6(4) = \min\{0,\ 4+3,\ 9-10\} = -1$, $pred(4) = 5$ (invariato).
- L'etichetta del nodo 4 diminuisce nella passata $n = 6$: $d^6 \neq d^5$, e un ciclo negativo è raggiungibile da 1.



Esempio: la tabella delle passate

Accanto a un'etichetta diminuita, tra parentesi il nuovo predecessore; in grassetto le voci lette all'indietro:

k	$d^k(1)$	$d^k(2)$	$d^k(3)$	$d^k(4)$	$d^k(5)$	$d^k(6)$
0	0	∞	∞	∞	∞	∞
1	0	7 (1)	2 (1)	∞	∞	∞
2	0	5 (3)	2	10 (2)	10 (3)	∞
3	0	5	2	0 (5)	10	15 (4)
4	0	5	1 (4)	0	10	5 (4)
5	0	4 (3)	1	0	9 (3)	5
6	0	4	1	-1 (5)	9	5

Esempio: lettura all'indietro dei predecessori

- Leggendo all'indietro dal nodo 4 (voci in grassetto): la riga 6 dà $pred = 5$, la riga 5 per il nodo 5 dà 3, la riga 4 per il nodo 3 dà 4, poi 5, 3, 1.

- La passeggiata è

$$1,\ 3,\ 5,\ 4,\ 3,\ 5,\ 4,$$

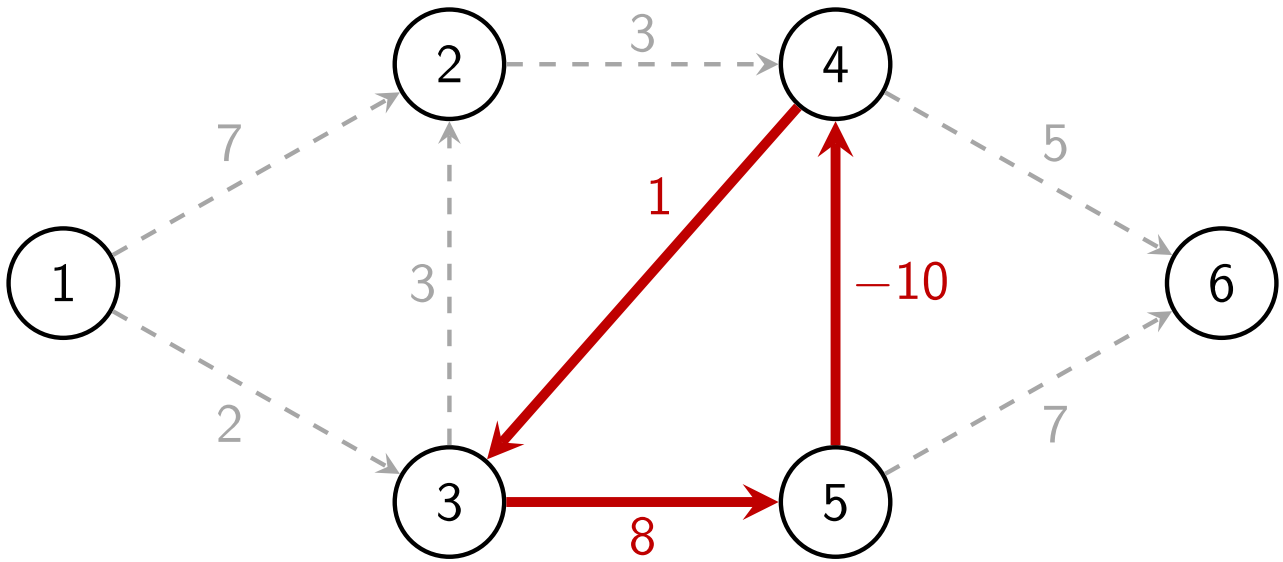
  di lunghezza

$$2 + 8 - 10 + 1 + 8 - 10 = -1 = d^6(4).$$

- Il nodo 3 si ripete: la passeggiata contiene il ciclo $3, 5, 4, 3$, di lunghezza -1.

Esempio: il ciclo negativo

Il ciclo $3, 5, 4, 3$ trovato dall'algoritmo, di lunghezza $8 - 10 + 1 = -1$ (in rosso):



Riepilogo

Le etichette d^k:

- $d^k(j)$ è la lunghezza di una passeggiata minima con al più k archi, calcolata passata per passata con una ricorsione
- Ogni passata esamina ogni arco una volta: tempo $O(n\,m)$ in totale

Arresto e rilevamento:

- Senza cicli negativi raggiungibili da s l'algoritmo si ferma entro n passate con le distanze
- Un cambiamento nella passata n rivela un ciclo negativo, trovato leggendo i predecessori all'indietro

L'algoritmo di Dijkstra

Introduzione: L'algoritmo di Dijkstra

- Quando tutte le lunghezze sono non negative, un algoritmo più veloce fissa un'etichetta alla volta
- L'algoritmo di Dijkstra è il metodo usato in pratica per le mappe stradali e le reti: tempo $O(n^2)$, o meno con un heap
- Dimostriamo perché funziona, e vediamo perché basta un solo arco negativo a farlo sbagliare

Nodi permanenti e temporanei

- In questa sezione tutte le lunghezze sono non negative: $d_{ij} \geq 0$ per ogni arco. Allora ogni ciclo ha lunghezza ≥ 0, e vale (A2).
- I nodi sono **permanenti**, quando la loro etichetta è definitiva, o **temporanei**.
- A ogni iterazione il nodo temporaneo con l'etichetta più piccola diventa permanente.

Algorithm: Algoritmo di Dijkstra

Input : un grafo orientato $G = (N, A)$ con lunghezze $d_{ij} \geq 0$, una sorgente $s \in N$

Output: le distanze $d(j) = \tilde{d}(j)$ e i predecessori $pred(j)$

1 **Passo 0 (inizializzazione).** Poni $d(s) := 0$, $d(j) := +\infty$ per $j \neq s$; tutti i nodi sono temporanei.

2 **Passo 1 (selezione).** Se nessun nodo temporaneo ha un'etichetta finita, `STOP`.

3 Altrimenti rendi permanente il nodo temporaneo i con il $d(i)$ più piccolo (quello di indice minimo, se ce n'è più di uno).

4 **Passo 2 (aggiornamento).** Per ogni arco $(i, j) \in \delta^+(i)$ con j temporaneo: se $d(i) + d_{ij} < d(j)$, poni $d(j) := d(i) + d_{ij}$ e $pred(j) := i$.

5 Torna al Passo 1.

Teorema 8: correttezza dell'algoritmo di Dijkstra

Se $d_{ij} \geq 0$ per ogni arco, a ogni iterazione dell'algoritmo di Dijkstra:

(a) *ogni nodo permanente i ha $d(i) = \tilde{d}(i)$;*

(b) *ogni nodo temporaneo j ha come $d(j)$ la lunghezza minima di un cammino da s a j i cui nodi diversi da j sono tutti permanenti ($+\infty$ se non ce n'è nessuno).*

Alla fine, $d(j) = \tilde{d}(j)$ per ogni $j \in N$.

Dimostrazione (1/2)

Entrambe le proprietà valgono dopo il Passo 0, senza nodi permanenti. Sia i il nodo scelto nel Passo 1.

1. $d(i) = \tilde{d}(i)$. Sia P un cammino qualsiasi da s a i, e sia h il suo primo nodo temporaneo. Il tratto di P fino a h ha lunghezza $\geq d(h)$ per (b), e il resto ha lunghezza ≥ 0, perché $d \geq 0$. Quindi

$$d(P) \geq d(h) \geq d(i),$$

e per (b) il valore $d(i)$ è la lunghezza di un cammino: $d(i) = \tilde{d}(i)$.

Dimostrazione (2/2)

2. *(b) dopo il Passo 2.* Sia j temporaneo e Q un cammino verso j i cui altri nodi sono permanenti, con h il nodo che precede j. Se $h = i$, allora $d(Q) \geq \tilde{d}(i) + d_{ij} = d(i) + d_{ij}$, il valore controllato nel Passo 2.

3. Se $h \neq i$, allora h è diventato permanente prima di i, attraverso un cammino di nodi permanenti di lunghezza $\tilde{d}(h)$; quando h è stato scelto, il Passo 2 ha dato $d(j) \leq \tilde{d}(h) + d_{hj} \leq d(Q)$.

Quindi il nuovo minimo è $\min\{d(j), d(i) + d_{ij}\}$, realizzato da un cammino, che è l'etichetta lasciata dal Passo 2. Alla fine i nodi temporanei hanno $d(j) = +\infty$: per (b) nessun arco va dall'insieme dei permanenti verso di loro, quindi non sono raggiungibili e $\tilde{d}(j) = +\infty$. □

Conseguenze

- Per il problema s–t l'algoritmo può fermarsi appena t diventa permanente, per (a).
- Gli archi $(pred(j), j)$ formano un albero dei cammini minimi: $pred(j)$ diventa permanente prima di j, e quando viene posto per l'ultima volta vale $d(j) = d(pred(j)) + d_{pred(j)\,j}$ con entrambe le etichette definitive.

Teorema 9: tempo di esecuzione dell'algoritmo di Dijkstra

Con le etichette in un array, l'algoritmo di Dijkstra richiede tempo $O(n^2)$.

Dimostrazione.

- Le iterazioni sono al più n, e il Passo 1 esamina le etichette temporanee in tempo $O(n)$.
- Il Passo 2 esamina la stella uscente del nodo scelto, e ogni nodo viene scelto una volta, quindi tutti gli aggiornamenti richiedono tempo $O(m)$.
- Il totale è $O(n^2 + m) = O(n^2)$, perché $m \leq n(n-1)$.

□

Con un heap

- Con le etichette temporanee in un heap binario, il Passo 1 e ogni diminuzione di un'etichetta richiedono tempo $O(\log n)$.
- Il totale è

$$O((n+m)\log n),$$

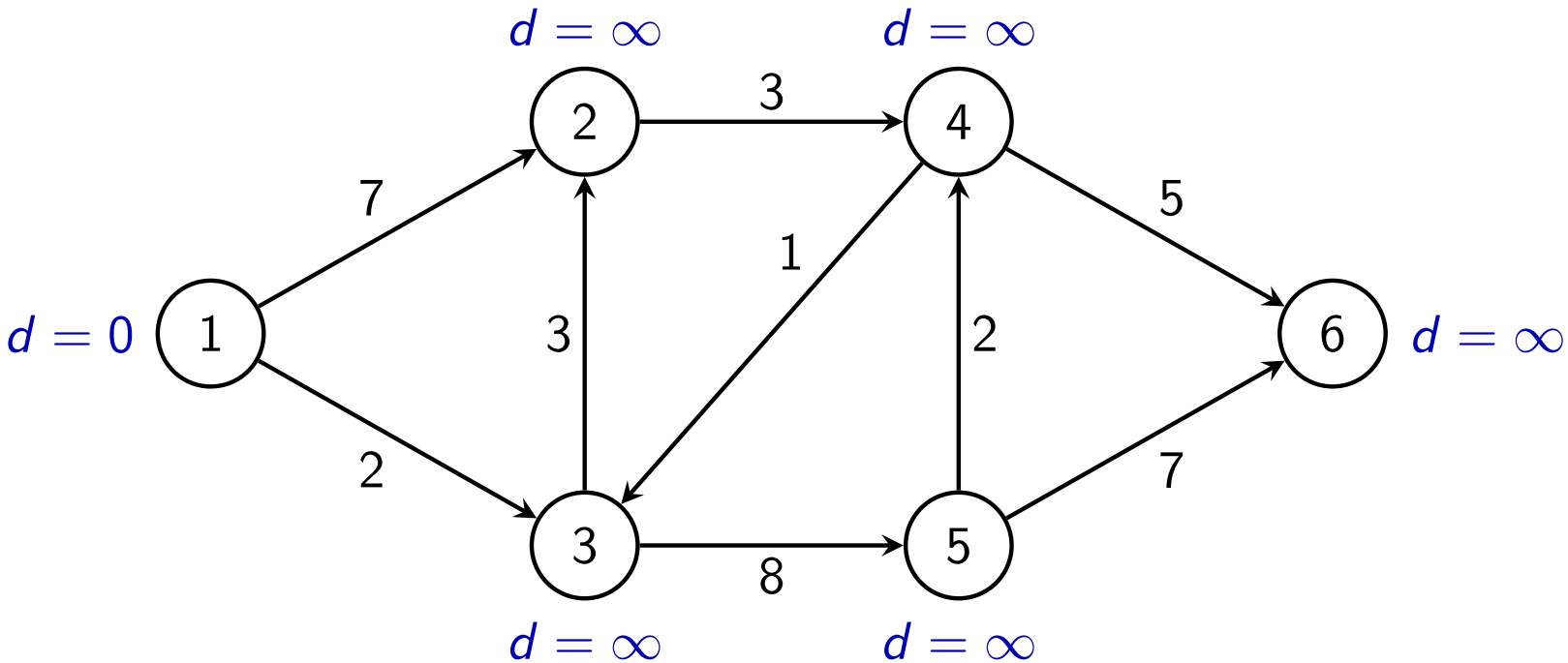
meglio di $O(n^2)$ sui grafi sparsi.

Esempio: Dijkstra dal nodo 1

- Eseguiamo l'algoritmo con le lunghezze del primo esempio.
- Nelle prossime slide il nodo scelto è in arancione, i nodi permanenti sono ombreggiati, un'etichetta appena diminuita è in rosso, e gli archi $(pred(j), j)$ sono in blu.

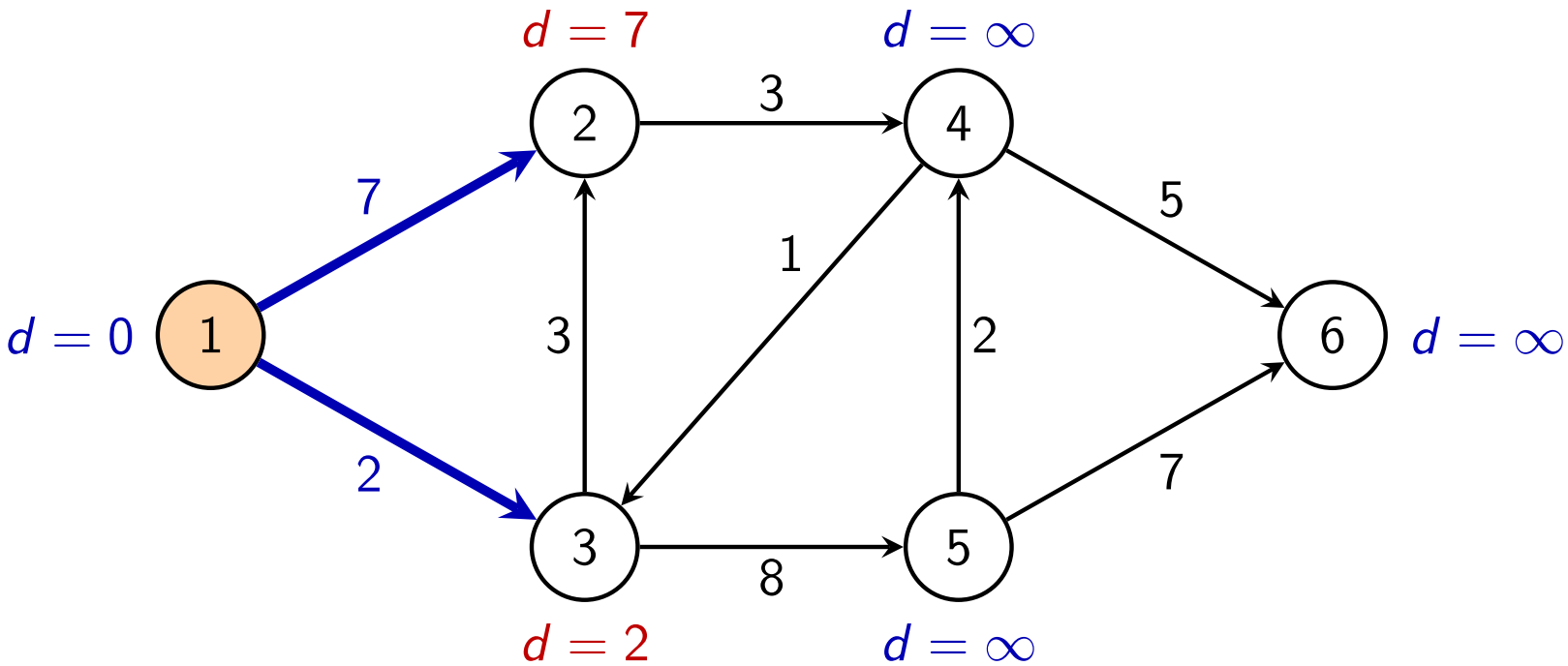
Esempio: Passo 0

- $d(1) = 0$, $d(j) = +\infty$ per $j \neq 1$; tutti i nodi sono temporanei.



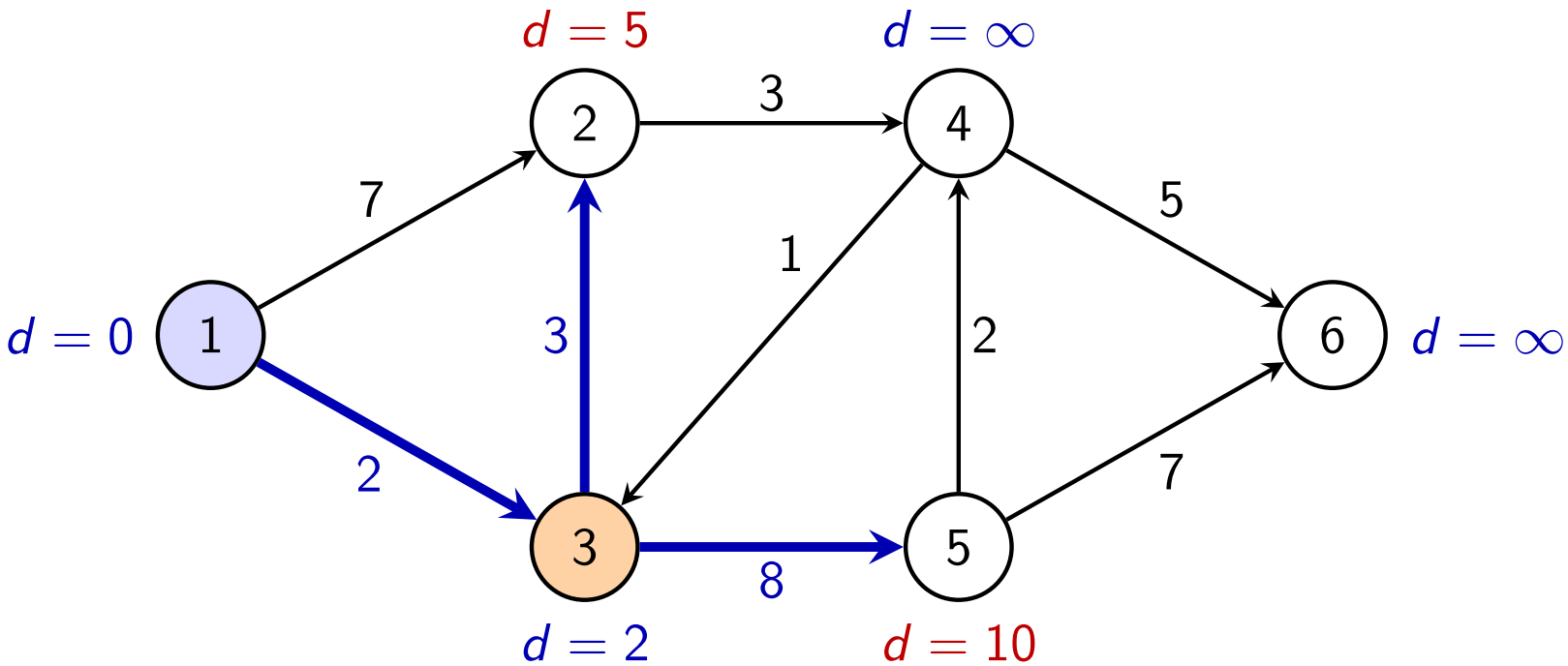
Esempio: iterazione 1

- Il nodo 1 ($d = 0$) diventa permanente.
- $(1, 2)$: $0 + 7 = 7 < \infty$, quindi $d(2) := 7$, $pred(2) := 1$;
- $(1, 3)$: $0 + 2 = 2 < \infty$, quindi $d(3) := 2$, $pred(3) := 1$.



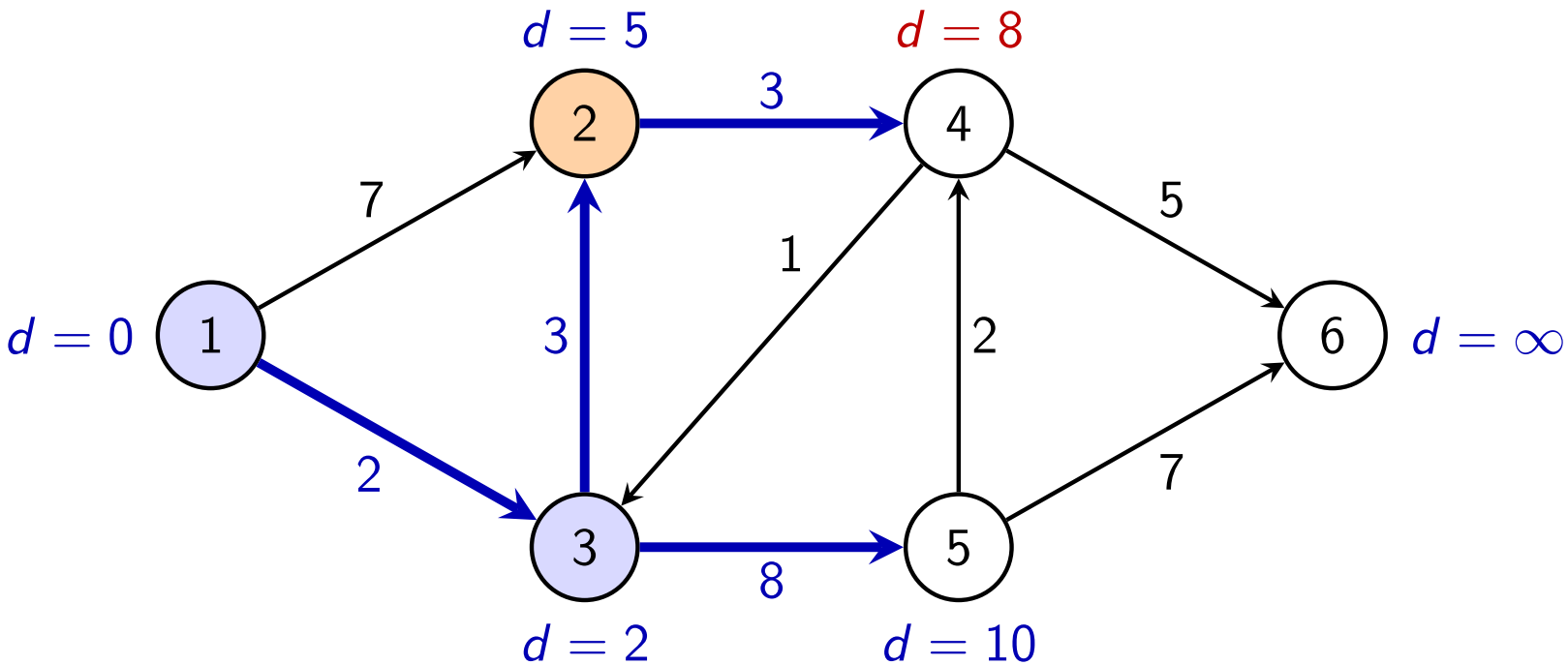
Esempio: iterazione 2

- Il nodo 3 ($d = 2$) diventa permanente.
- $(3, 2)$: $2 + 3 = 5 < 7$, quindi $d(2) := 5$, $pred(2) := 3$;
- $(3, 5)$: $2 + 8 = 10 < \infty$, quindi $d(5) := 10$, $pred(5) := 3$.



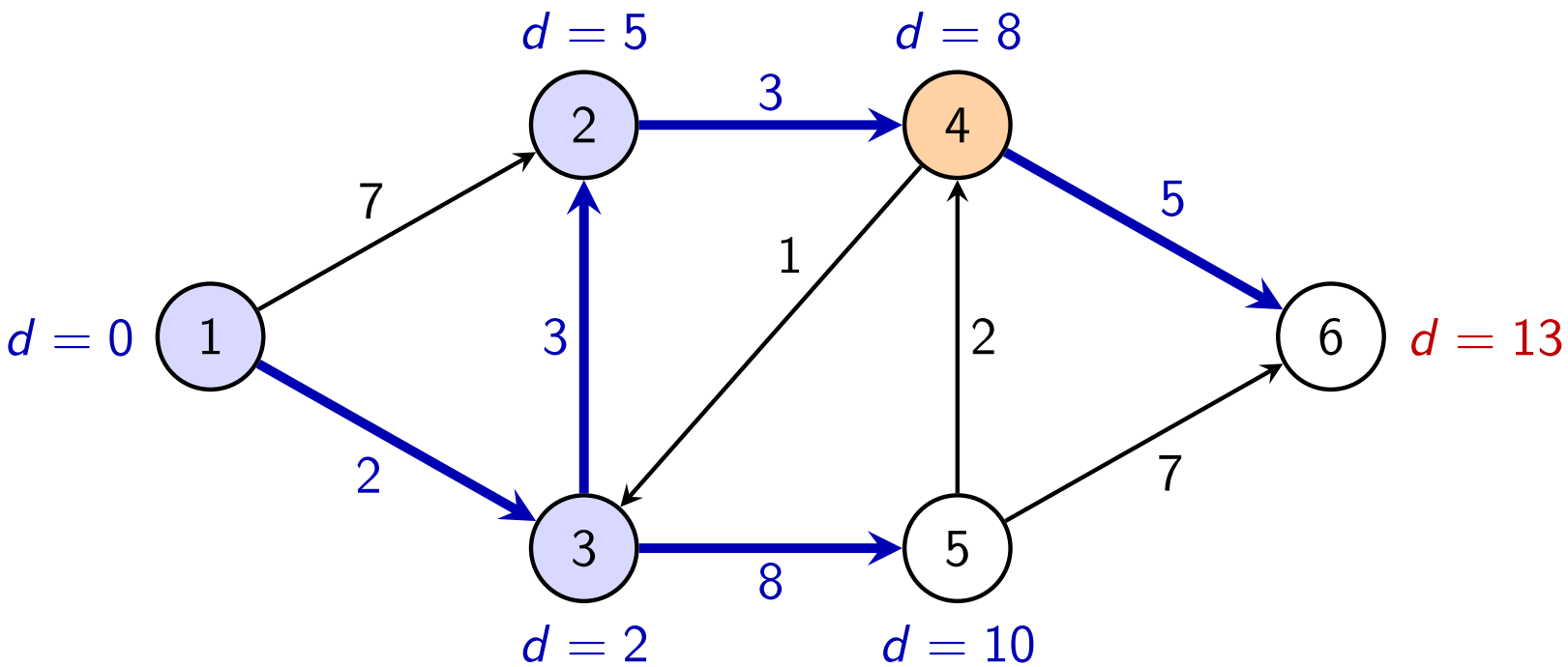
Esempio: iterazione 3

- Il nodo 2 ($d = 5$) diventa permanente.
- $(2, 4)$: $5 + 3 = 8 < \infty$, quindi $d(4) := 8$, $pred(4) := 2$.



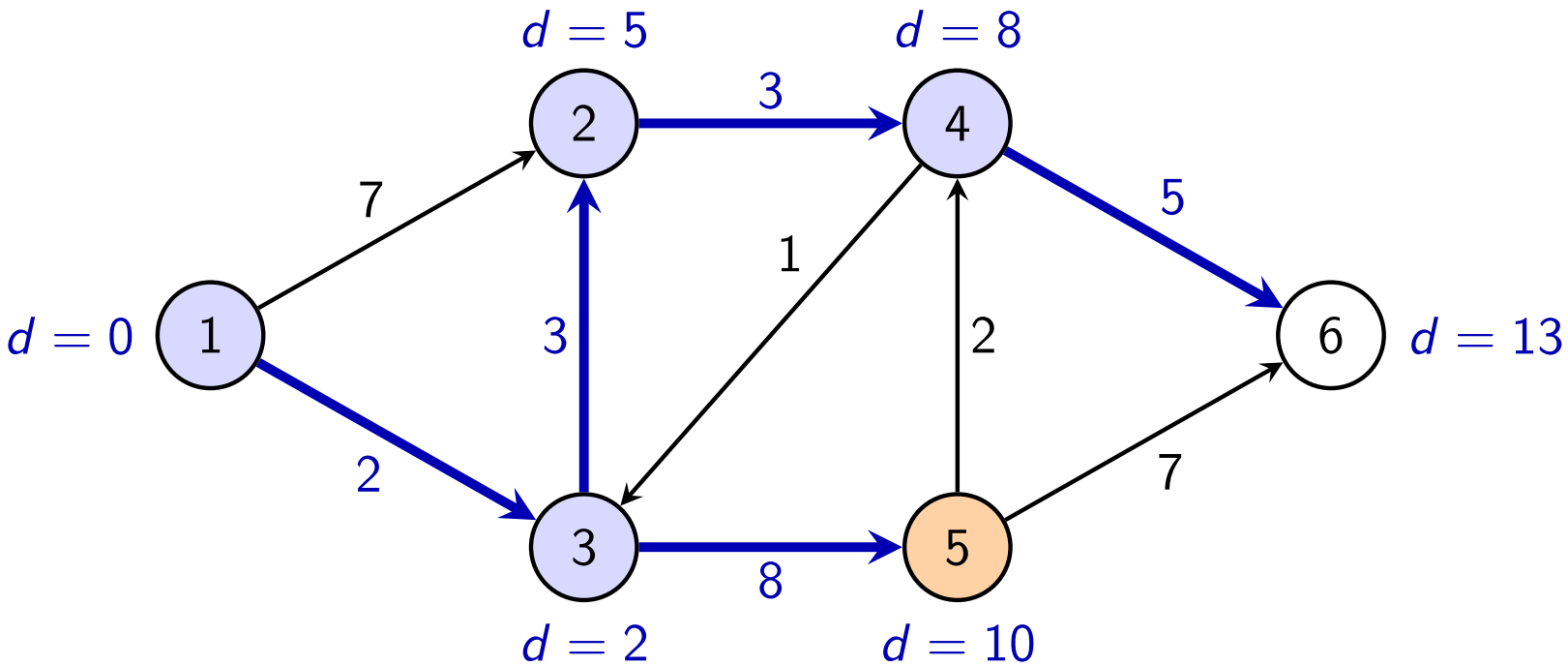
Esempio: iterazione 4

- Il nodo 4 ($d = 8 < 10$) diventa permanente; $(4, 3)$ termina in un nodo permanente.
- $(4, 6)$: $8 + 5 = 13 < \infty$, quindi $d(6) := 13$, $pred(6) := 4$.



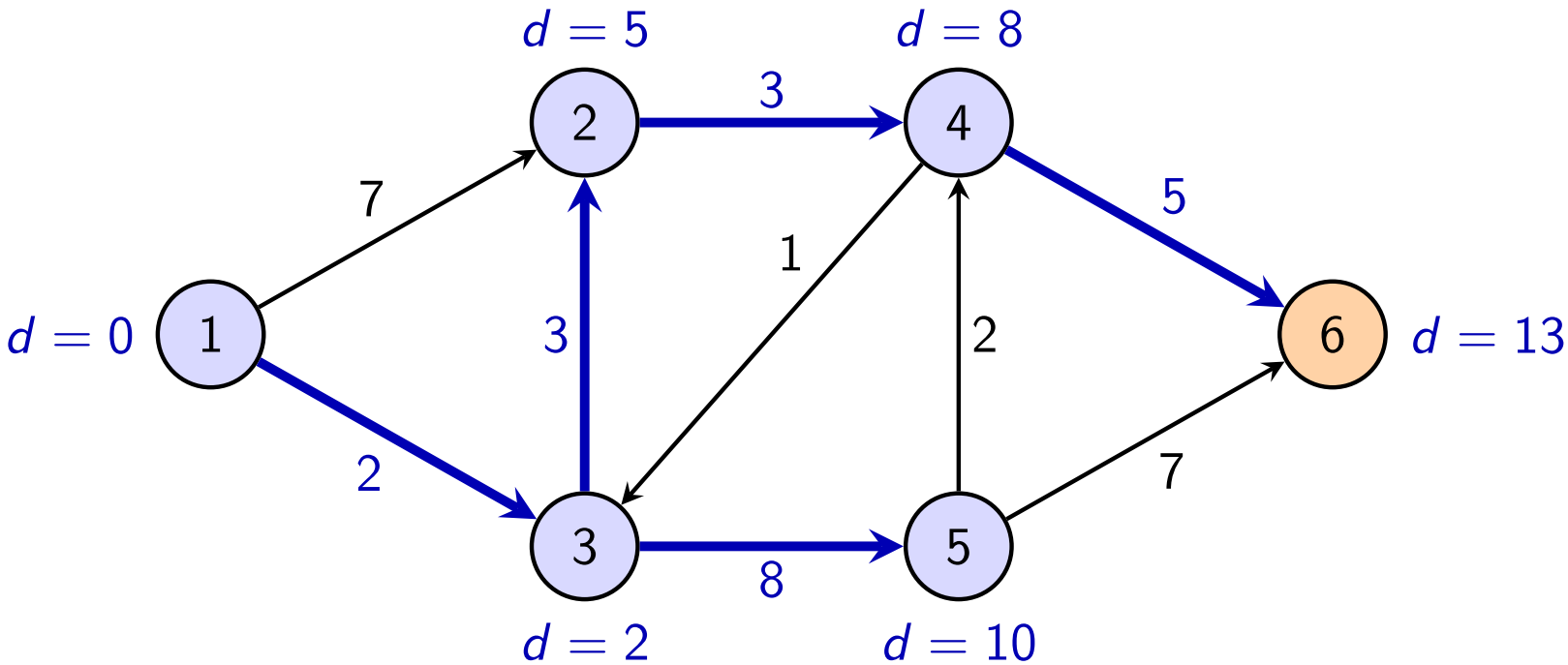
Esempio: iterazione 5

- Il nodo 5 ($d = 10$) diventa permanente. L'arco $(5, 6)$ offre $10 + 7 = 17 > 13$, e l'arco $(5, 4)$ termina in un nodo permanente: nessun cambiamento.



Esempio: iterazione 6

- Il nodo 6 ($d = 13$) diventa permanente; non ha archi uscenti. Non resta alcun nodo temporaneo: `STOP`.



Esempio: la tabella delle iterazioni

Ogni riga è un'iterazione: il nodo scelto diventa permanente (etichetta in grassetto), e un'etichetta diminuita ha il nuovo predecessore tra parentesi.

iterazione	nodo scelto i	$d(1)$	$d(2)$	$d(3)$	$d(4)$	$d(5)$	$d(6)$
1	1	0	7 (1)	2 (1)	∞	∞	∞
2	3	0	5 (3)	2	∞	10 (3)	∞
3	2	0	5	2	8 (2)	10	∞
4	4	0	5	2	8	10	13 (4)
5	5	0	5	2	8	10	13
6	6	0	5	2	8	10	13

Esempio: il risultato

- Nell'iterazione 2 l'etichetta del nodo 2 scende da 7 a $2+3=5$; nell'iterazione 5 l'arco $(5,6)$ offre $10+7=17>13$, e l'arco $(5,4)$ termina in un nodo permanente.

- I nodi diventano permanenti nell'ordine $1,3,2,4,5,6$, con

$$\tilde{d}=(0,5,2,8,10,13);$$

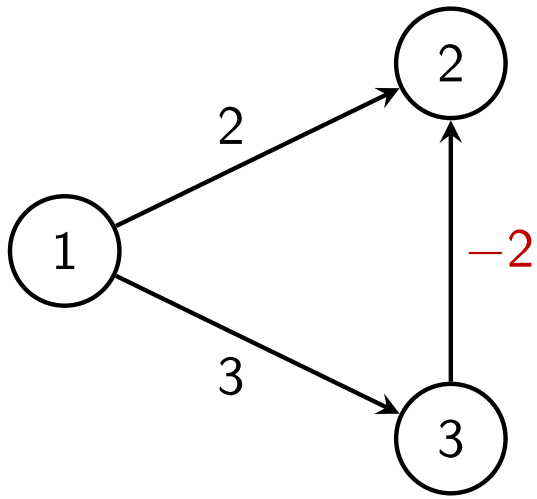
i predecessori danno l'albero dei cammini minimi trovato con le equazioni di Bellman.

Perché serve $d_{ij} \geq 0$?

Basta un solo arco negativo a far sbagliare l'algoritmo di Dijkstra.

Esempio (1/3)

Nel grafo qui sotto, con $s = 1$, l'algoritmo rende permanente il nodo 2 con $d(2) = 2$, poi il nodo 3 con $d(3) = 3$; l'arco $(3, 2)$ termina in un nodo permanente e viene ignorato. Il cammino $1, 3, 2$ ha lunghezza $3 - 2 = 1 < 2$.



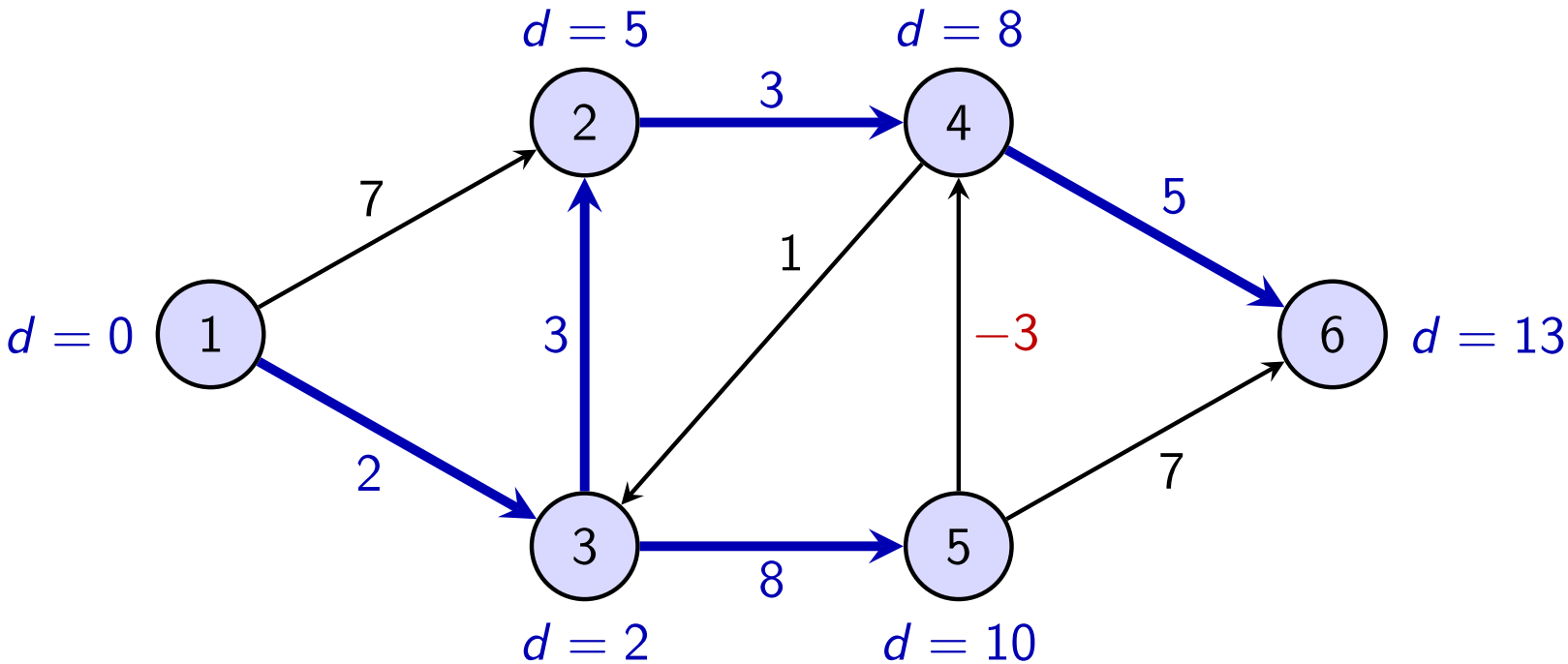
Esempio (2/3)

- Sul grafo di riferimento con le lunghezze d' dell'esempio di Bellman–Ford ($d'_{54} = -3$), l'algoritmo di Dijkstra rende permanenti i nodi nell'ordine $1, 3, 2, 4, 5, 6$ e restituisce $d(4) = 8$ e $d(6) = 13$.

- Il nodo 4 diventa permanente prima del nodo 5, quindi il cammino più corto $1, 3, 5, 4$, di lunghezza 7, viene perso, e con esso il cammino verso 6 di lunghezza 12.

Esempio (3/3)

Le etichette restituite dall'algoritmo di Dijkstra con $d'_{54} = -3$: quando il nodo 5 diventa permanente, l'arco $(5, 4)$ termina in un nodo permanente e viene ignorato.



Dove la dimostrazione non regge

- La dimostrazione della correttezza dell'algoritmo di Dijkstra usa $d \geq 0$ in un solo punto: il resto di P dopo il suo primo nodo temporaneo ha lunghezza ≥ 0.
- Un arco negativo fa cadere proprio questo passaggio.
- Con un arco negativo si usa invece l'algoritmo di Bellman–Ford, oppure l'ordine topologico se il grafo è aciclico.

Riepilogo

L'algoritmo di Dijkstra:

- Con lunghezze non negative rende permanente, a ogni iterazione, il nodo temporaneo con l'etichetta più piccola, la cui etichetta è allora definitiva
- Richiede tempo $O(n^2)$, $O((n+m)\log n)$ con un heap

Lunghezze negative:

- Con un arco negativo può sbagliare
- Si usa invece l'algoritmo di Bellman–Ford

Errori tipici

Introduzione: Errori tipici

- Raccogliamo gli errori che si fanno più spesso con i cammini minimi
- Riguardano la scelta dell'algoritmo, il segno dei potenziali e i test di arresto
- Controllali prima di eseguire un algoritmo a mano

Errori tipici (1/4)

Osservazione 2: Eseguire l'algoritmo di Dijkstra con un arco negativo

Le sue etichette possono essere sbagliate, come con $d'_{54} = -3$ sul grafo di riferimento; si usa Bellman–Ford, o l'ordine topologico se il grafo è aciclico.

Osservazione 3: Aggiungere una costante a ogni lunghezza

Rende le lunghezze non negative ma cambia i cammini minimi: un cammino con k archi cresce di k volte la costante.

Errori tipici (2/4)

Osservazione 4: Scambiare il segno dei potenziali

Con $+1$ nella coda di $\boldsymbol{B}$ i vincoli duali sono $\pi_i - \pi_j \leq d_{ij}$ e i potenziali ottimi sono $\pi_i = -\tilde{d}(i)$.

Osservazione 5: Saltare il test della passata n

Dopo $n-1$ passate le etichette sono le distanze solo in assenza di cicli negativi; un cambiamento nella passata n ne rivela uno.

Errori tipici (3/4)

Osservazione 6: Aspettarsi che Bellman–Ford veda ogni ciclo negativo

Rileva solo i cicli raggiungibili da s, mentre un ciclo negativo in un punto qualsiasi di G rende illimitato il modello LP.

Osservazione 7: Aggiornare un nodo permanente nell'algoritmo di Dijkstra

La sua etichetta è definitiva; si aggiornano solo i nodi temporanei.

Errori tipici (4/4)

Osservazione 8: Cercare un ordine topologico in un grafo con cicli

Esiste solo nei grafi aciclici: il grafo di riferimento non ne ha.

Riepilogo

Scegliere l'algoritmo:

- Lunghezze non negative: Dijkstra; grafo aciclico: ordine topologico; altrimenti Bellman–Ford
- Mai spostare le lunghezze di una costante

Eseguirlo:

- Bellman–Ford: fare sempre il test della passata n; vede solo i cicli raggiungibili da s
- Dijkstra: le etichette permanenti non si aggiornano mai; potenziali: $\pi_i = -\tilde{d}(i)$

In sintesi

Introduzione: In sintesi

- Raccogliamo le idee chiave del capitolo sui cammini minimi
- L'ultima slide riunisce i risultati sul grafo di riferimento, per tutte le lunghezze usate nel capitolo

Riepilogo (1/3)

- La distanza $\tilde{d}(j)$ è la lunghezza di un cammino minimo da s a j; è ben definita quando nessun ciclo negativo è raggiungibile da s (A2).
- I sottocammini dei cammini minimi sono minimi; le distanze soddisfano le equazioni di Bellman, ed esiste un albero dei cammini minimi.
- Potenziali ammissibili, $\bar{c}_{ij} = d_{ij} - \pi_i + \pi_j \geq 0$, e un cammino di archi stretti certificano l'ottimalità.

Riepilogo (2/3)

- Il modello LP è un flusso unitario, $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq 0$; il suo duale è
$$\max\{\pi_s - \pi_t : \pi_i - \pi_j \leq d_{ij}\},$$
ottimo in $\pi_i = -\tilde{d}(i)$.
- Grafi aciclici: ordine topologico, $O(n+m)$, lunghezze qualsiasi.
- Bellman–Ford: $O(n\,m)$, lunghezze qualsiasi, rileva i cicli negativi.
- Dijkstra: $O(n^2)$, lunghezze non negative.

Riepilogo (3/3)

Il grafo di riferimento in sintesi.

lunghezze d (Dijkstra)	$\tilde{d} = (0, 5, 2, 8, 10, 13)$, ordine $1, 3, 2, 4, 5, 6$
modelli LP	$z(\mathrm{SP}(1, 6)) = 13$, $z(\mathrm{SP}(1)) = 38$
lunghezze d', $d'_{54} = -3$ (Bellman–Ford)	$\tilde{d} = (0, 5, 2, 7, 10, 12)$, 5 passate
lunghezze d'', $d''_{54} = -10$	ciclo negativo $3, 5, 4, 3$ di lunghezza -1
grafo aciclico, arco $(3, 4)$	ordine $1, 3, 2, 5, 4, 6$; $\tilde{d} = (0, 5, 2, 3, 10, 8)$