Estrategias algorítmicas

Backtracking

Algoritmos codiciosos

Programación dinámica

Estrategias algorítmicas

Backtracking

Algoritmos codiciosos

Programación dinámica

Tres problemas en grafos *con costos*

a) Grafos direccionales:

- encontrar la *ruta más corta desde un vértice a todos los otros* —el algoritmo de Dijkstra

b) Grafos no direccionales:

- **encontrar el *árbol de cobertura de costo mínimo***

c) Grafos direccionales:

- encontrar las *rutas más cortas entre todos los pares de vértices*

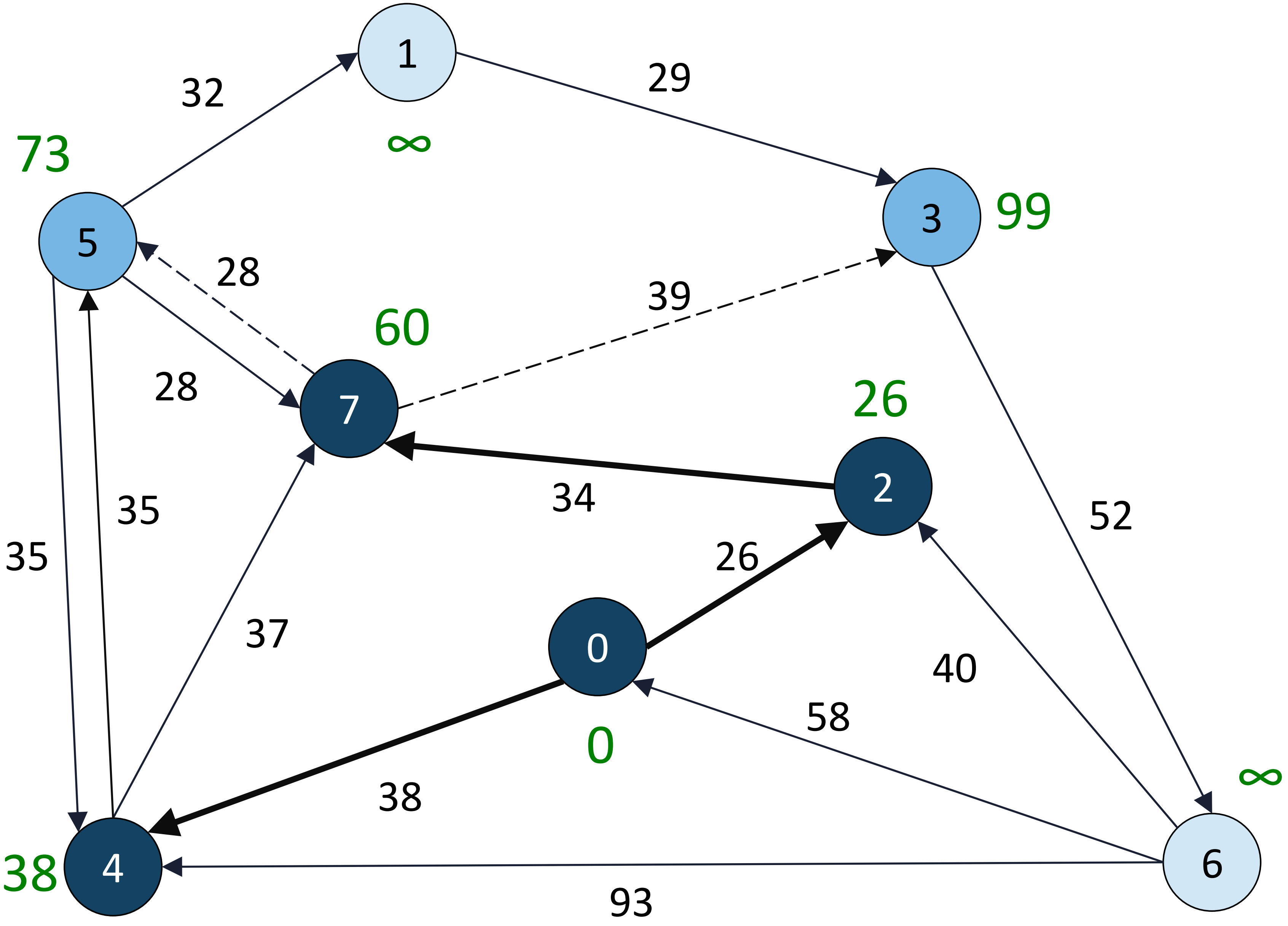
El algoritmo de Dijkstra es un **algoritmo codicioso**

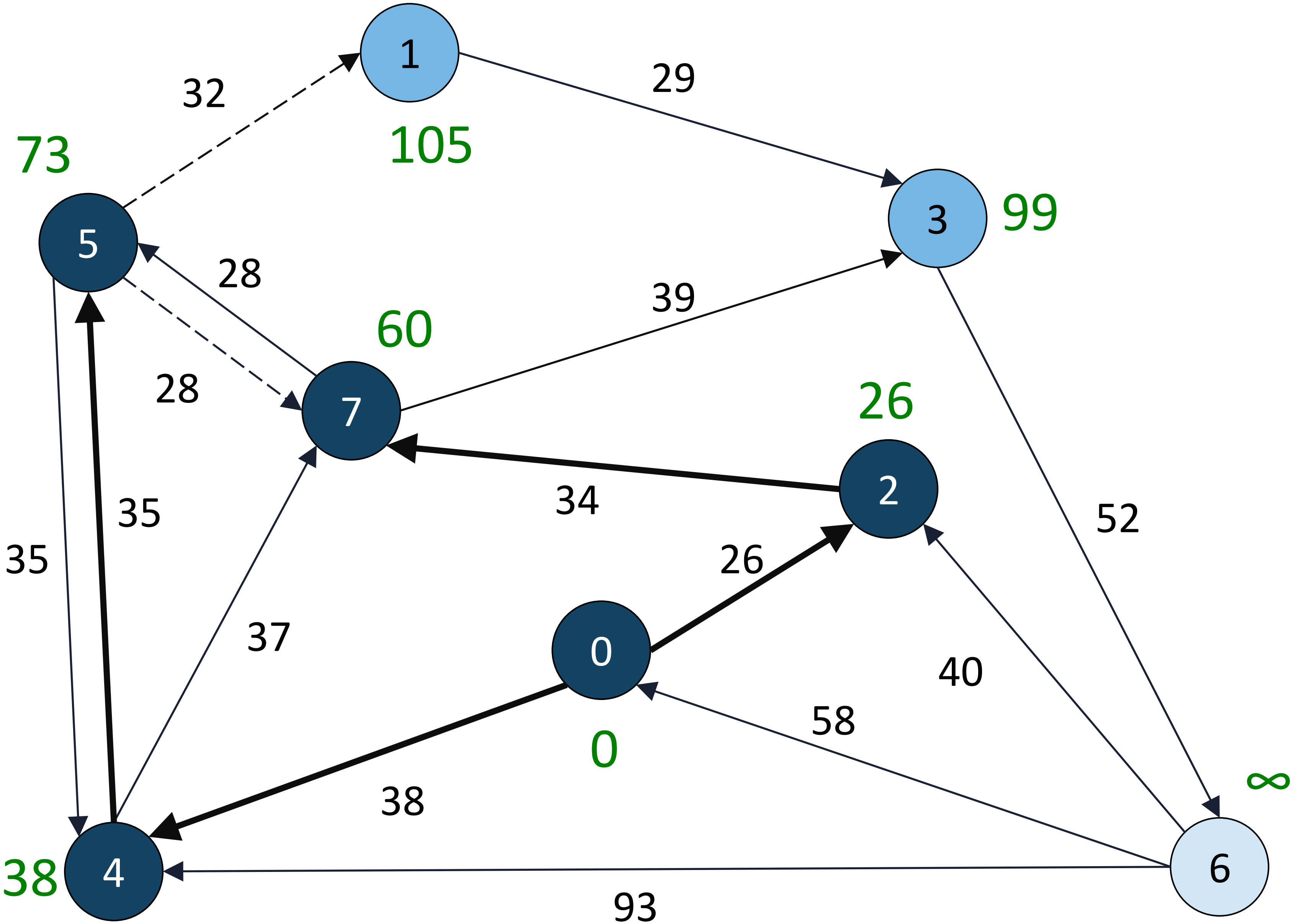
En cada paso, el algoritmo hace una elección que corresponde a un **óptimo local**:

- toma la mejor decisión que puede, con la información que tiene hasta ese momento

… con la esperanza de que al hacer la última elección haya logrado el (un) **óptimo global:**

- algunos algoritmos codiciosos efectivamente encuentran el óptimo global (p.ej., Dijkstra)
- … pero no todos


1
∞
32
29
73
5
3
99
28
39
60
28
7
26
2
34
35
35
52
26
37
0
40
58
0
38
∞
6
38
4
93

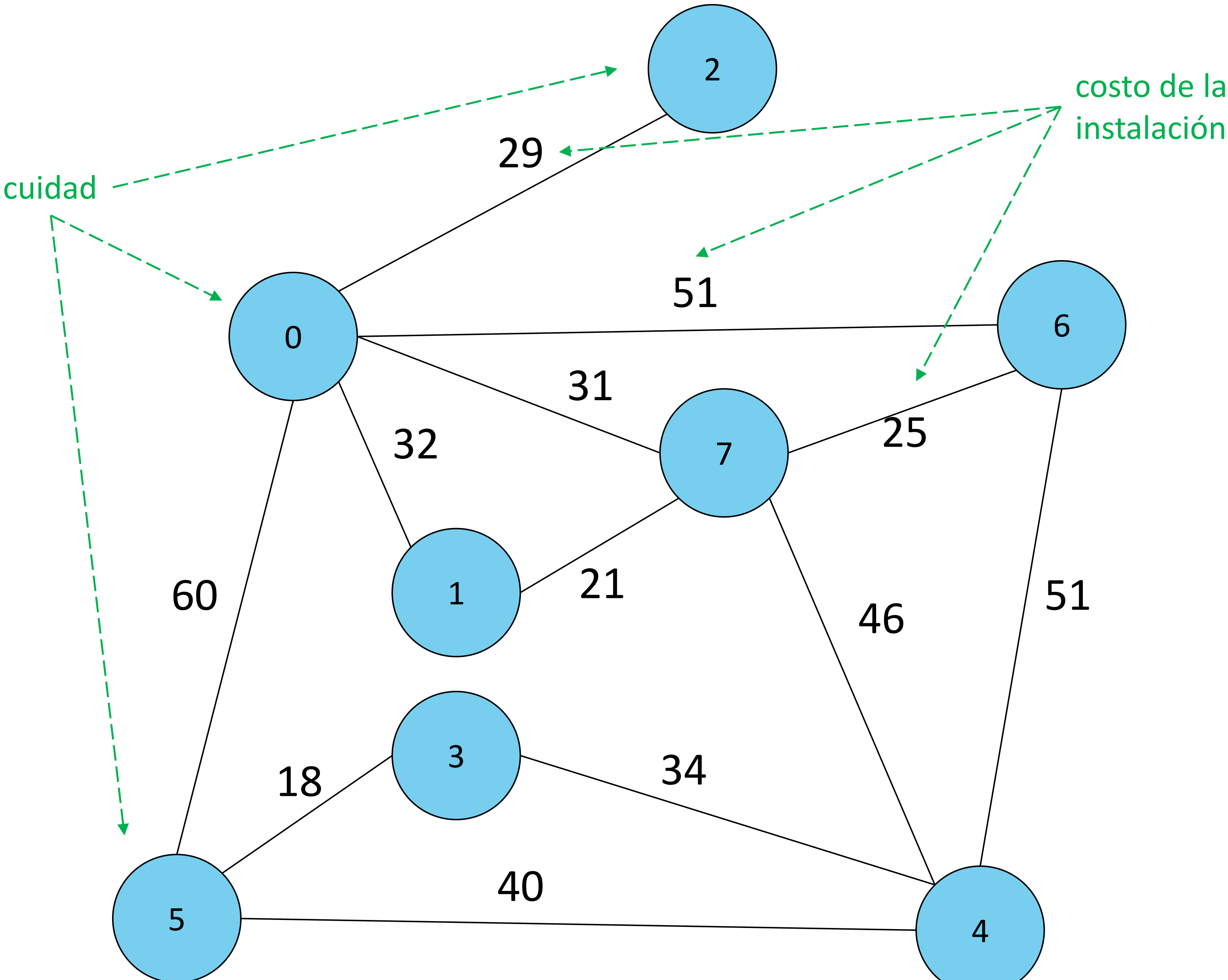

1
105
29
32
73
5
3
99
28
39
60
28
7
26
2
34
35
35
26
0
37
40
58
0
∞
38
38
4
6
93

Conectividad digital

- El intendente de la Región del Maule ha decidido mejorar significativamente la conectividad digital de la región
- La idea es instalar fibra óptica subterránea entre todos los pares de puntos relevantes de la región —cada instalación tiene un costo
- Sólo que hay demasiada fibra óptica que instalar como para hacerlo todo de una vez
- Lo prioritario es conectar las ciudades más pobladas, que tienen escuelas, universidades, hospitales, compañías de bomberos, super-mercados, etc.

¿Cuál es la forma más barata de hacer esto?

VI REGION DEL LIB. GRAL. BERNARDO O'HIGGINS
VII REGION DEL MAULE
Bucalemu
Lolol
El Guaico
Chépica
Convento Viejo
Auquinco
Quinta
Codegua
La Rufina
Vn Tinguiririca 4300
Boyeruca
Lo Valdivia
Morza
Huemul
Sierra Bellavista
Trompetilla
Termas del Flaco
RESERVA NACIONAL LAGUNA TORCA
Llico
San Pedro de Alcántara
Comalle
Teno
Rebeca
Río Claro
Río Las Damas
Ranguil
Est Las Palmas
Lago Vichuquén
Río Teno
Co Cayetano 2886
El Manzano
Lipimávida
Rauco
La Montaña
Culenar
Los Queñes
Pichibudi
Vichuquén
Romeral
Pta Duao
Est Nilahue
Trincao
CURICO
Cerro Torre de Santa Elena 3820
Las Leñas
Los Molles
Hualañé
Potrero Grande Chico
Licantén
Palquibudi
Sagrada Familia
Iloca
Lontué
Los Niches
Upeo
Monte Oscuro
El Planchón
Paso de Vergara 2502
Laguna Teno
60 Kms a Malargüe
La Lora
Villa Prat
Itahue
Molina
Río Lontué
Río Mataquito
Potrero Grande
Quebrada Honda
Vn Peteroa 4151
Baños de Azufre
35°
Curepto
Yacel
Río Claro
La Trinchera
Culenar
Paso de Potrerillos 2908
Río Grande
Aduana
Cumpeo
Río Colorado
Océano Pacífico
Mar Chileno
Coipué
Gualleco
San Rafael
Río Patos de San Pedro
Putú
Botalcura
Astillero
Radal
RESERVA NACIONAL RADAL SIETE TAZAS
Junquillar
Batuco
Pelarco
Vn Descabezado Chico 3750
Quivolgo
Balseo
Río Huenchullamí
TALCA
Río Lircay
Constitución
Pencahue
Vn Descabezado Grande 3830
CHILE
ARGENTINA
Curtiduría
Corinto
Aurora
RESERVA NACIONAL ALTOS DEL LIRCAY
Colín
Río Maule
Pichamán
San Clemente
Corralones
Vilches
Maule
Corel
149 Kms a Malargüe
Duao
Las Cañas
Nirivilo
Co Nirivilo 722
Hotel Casas El Colorado
Cta Pellines
San Javier
Santa Ana
Río Maule
Puente Pardo
Río Loanco
Río Purapel
Mingre
Posada Paso Nevado
Armerillo
Río de la Invernada
Laguna La Invernada
Empedrado
Huerta del Maule
Colbún
Lago Colbún
BARRERA
Villa Alegre
Lago Machicura
Paso Troión 3044
Pahuil
Las Campanas
Yerbas Buenas
RESERVA NACIONAL LOS BELLOTOS
Las Garzas
Endesa
La Mina
Río Puelche
Río Reloca
Melozal
Panimávida
Rari
Baños de La Monja
Quinamávida
Los Rabones
Melado
Arbolillo
Túnel Canal Melado
RESERVA NACIONAL FEDERICO ALBERT
La Aguada
Palmilla
LINARES
Roblería
El Salto
Paso Pehuenche o del Maule 2553
Chanco
Tapihue
Río Ancoa
36°
Pueblo Hundido
Villa Seca
Miraflores
Campamento Ancoa
RESERVA NACIONAL LOS RUILES
Pelluhue
Llepo
Co El Toro 3085
Vn San Pedro 3621
Lgna del Maule
Mesamávida
Curanipe
Los Nabos
Longaví
Río Melado
Los Perales
Melado
Lgna Negra
Pya Pelluhue
Río de la Puente
Aduana Pejerrey
Los Cristales
CAUQUENES
Paso Hondo
Retiro
Co El Buitre 2955
Pta Nugurne
Hualve
Quella
Unicavén
El Tránsito
Tregualemu
RESERVA NACIONAL LOS QUEULES
El Peñón
Río Achibueno
Nvdo Longaví 3242
Lgna Fea
Río Longaví
Pullay
Coronel
Quinchimávida
Parral
Ajial
SIERRA DE COCHICO
Río Barrancas
VIII REGION DEL BIOBIO
VII REGION DEL MAULE
Cancha Alegre
Tres Esquinas
Villa Rosa
San Pablo
Laguna Varvarco Campos
Buchupureo
51 Kms a Quirihue
60 Kms a Chillán
72°
71°
El Parrón
Pocillas
Emb Digua
Bullileo
Cobquecura
Perquilauquén
Termas de Catillo
San Gregorio
Quirihue
Lago Dial
Colmuyao
Torrecilla
San Carlos
Zemita
Río Perquilauquén
Ninhue
Cachapoal
Mela
El Sauce
BALSEO
Vn Domuyo 4709
Trehuaco
Coipín
Nahueltoro
San Fabián de Alicó
Vegas de Itata
San Nicolás
Río Los Sauces

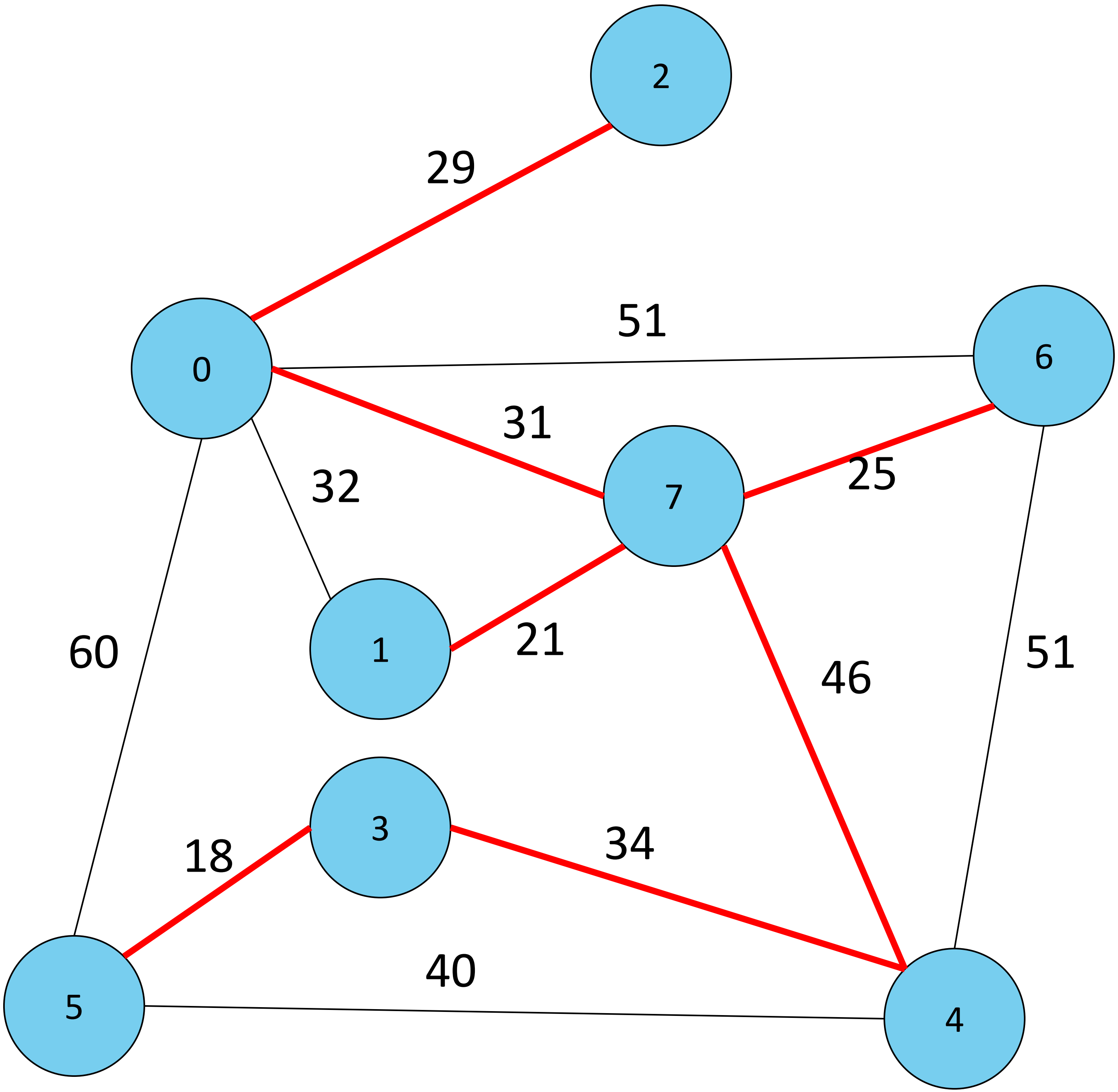

2
costo de la
instalación
29
cuidad
51
0
6
31
7
32
25
1
21
60
46
51
3
18
34
40
5
4

MST: *Minimum spanning tree*

El problema es descrito por un **grafo no direccional con costos**

La solución es un **subconjunto *T* de las aristas** tal que:

- las aristas de *T* no forman ciclos —forman un **árbol**
- las aristas de *T* incluyen todos los vértices —forman una **cobertura**
- no hay otro árbol de cobertura con menor costo total (la suma de los costos de las aristas de *T*) —es de **costo mínimo**


2
29
0
51
6
31
7
32
25
1
21
60
46
51
3
18
34
40
5
4

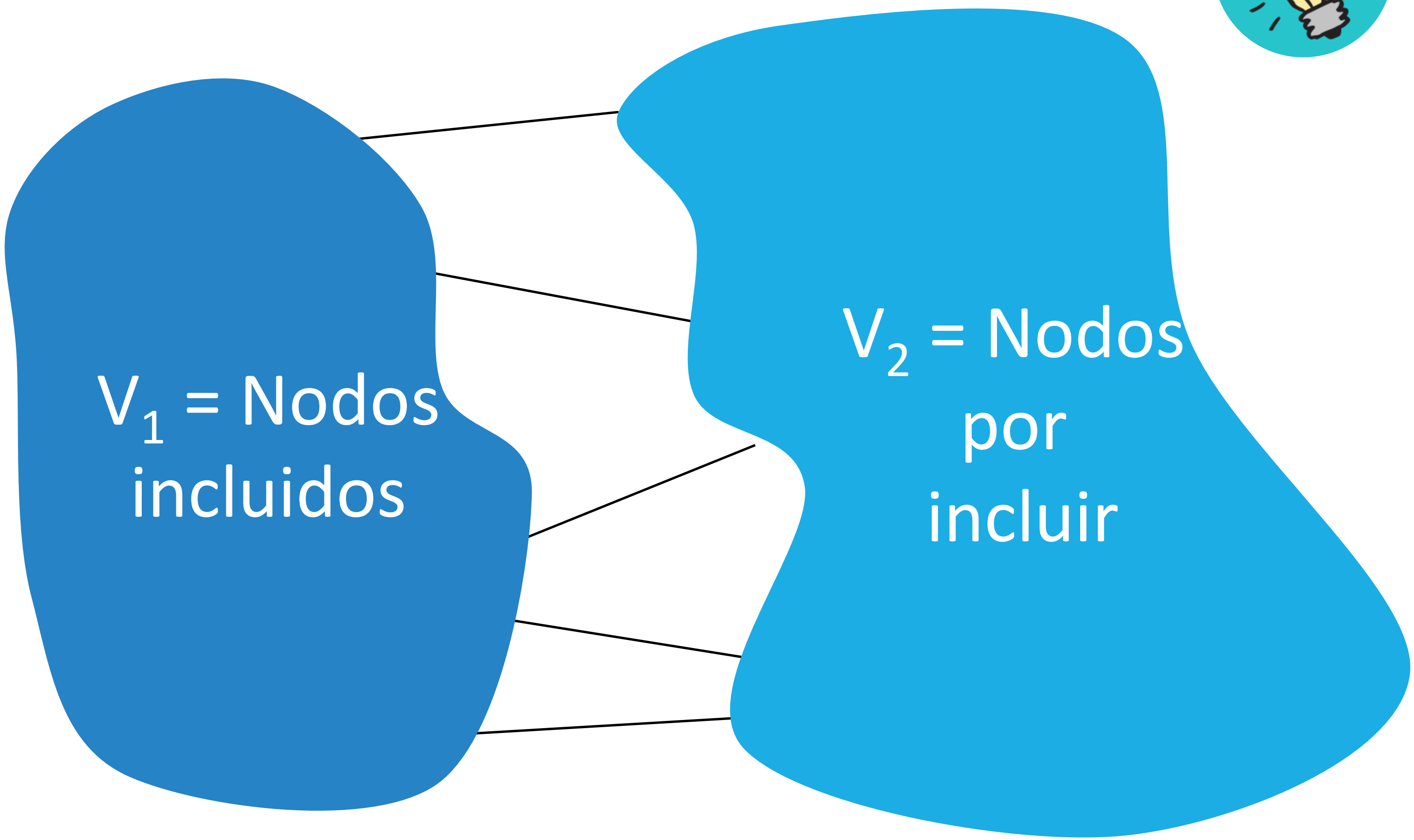
MSTs, **cortes** y aristas que **cruzan** el corte

Cortemos (particionemos) el grafo en dos (sub)conjuntos de vértices, V_1 y V_2

Una arista **cruza** el corte si un extremo está en V_1 y el otro en V_2

¿Qué podemos afirmar respecto a estas aristas y los **MST**?

El corte (V_1, V_2) y las aristas que cruzan el corte



¿Cuál debería ser el siguiente nodo a incluir?

Buscando un MST

Si para cada corte la arista de menor costo está en un **MST**

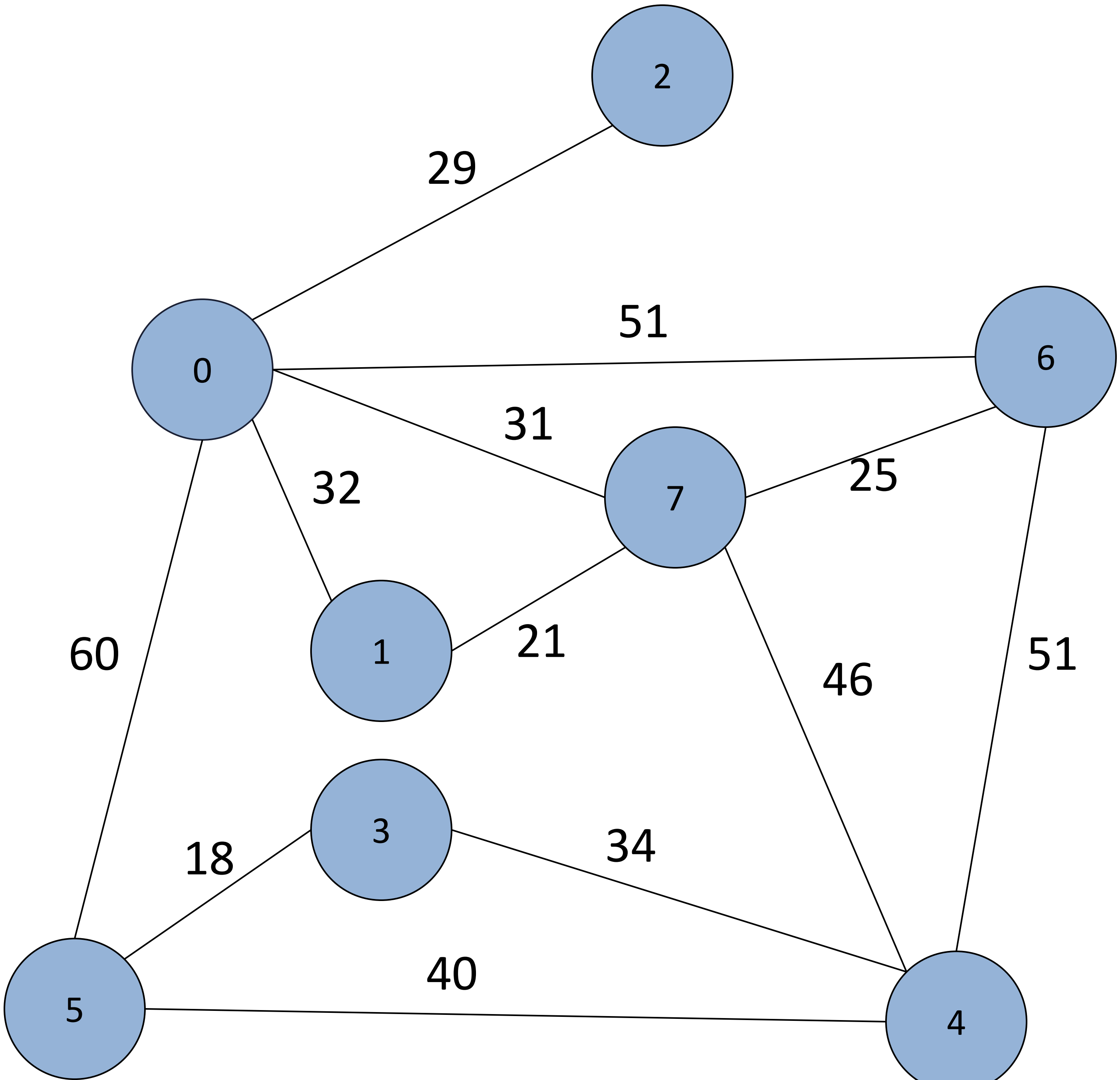
… ¿cómo podemos encontrar un **MST**?

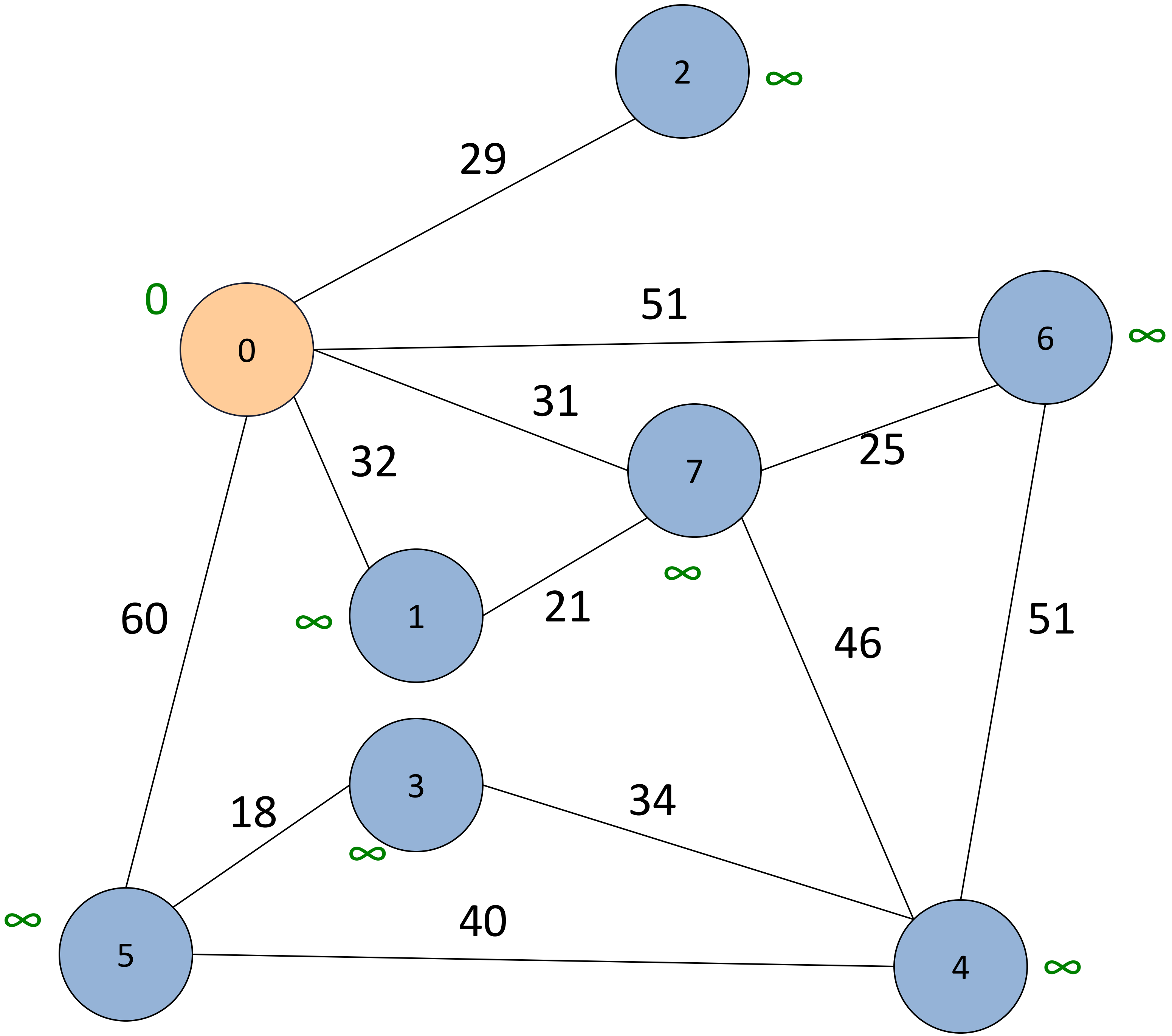
… ¿podremos construirlo una arista a la vez?

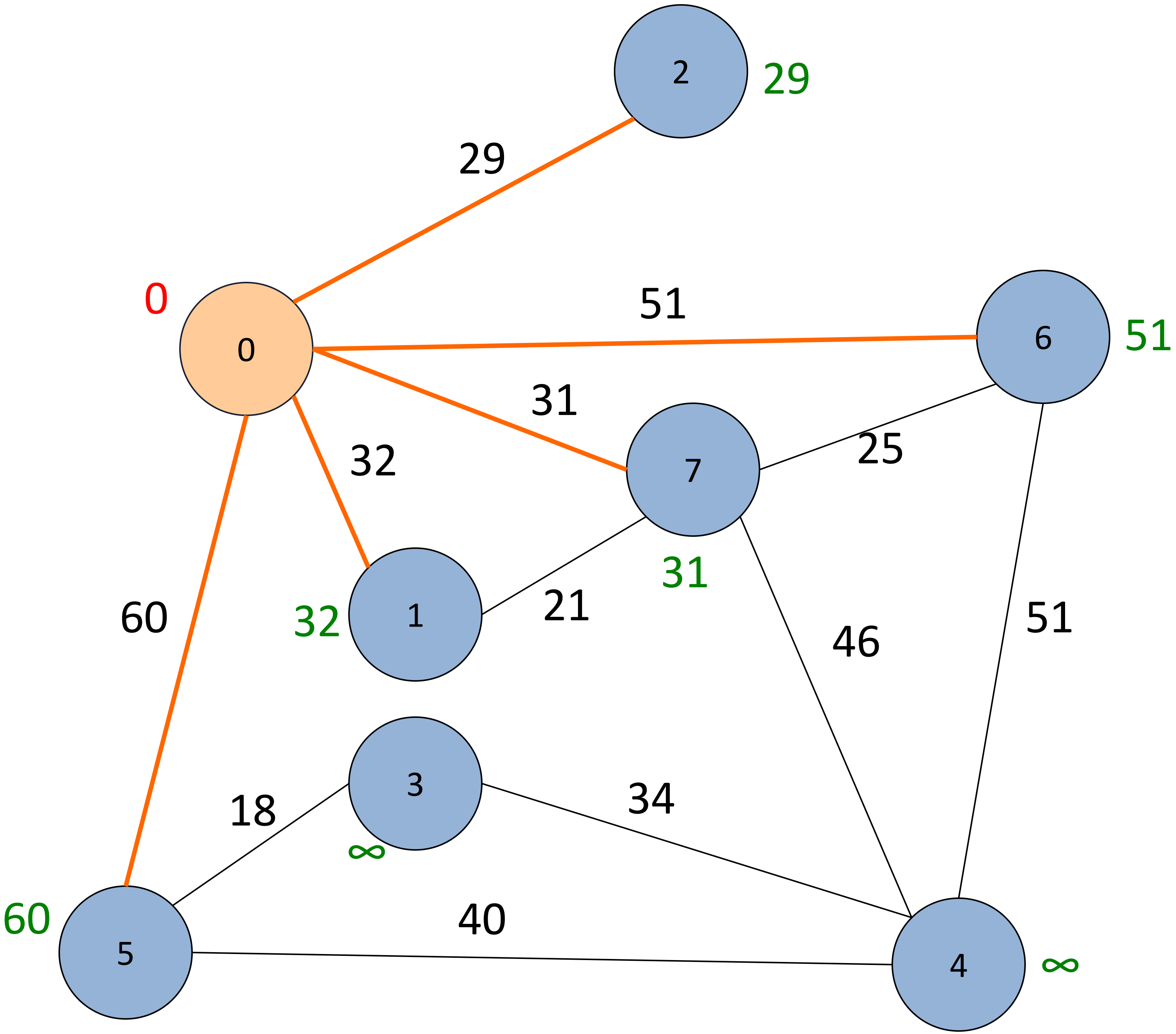
Algoritmo de Prim

Para un grafo $G(V, E)$, y un nodo inicial x

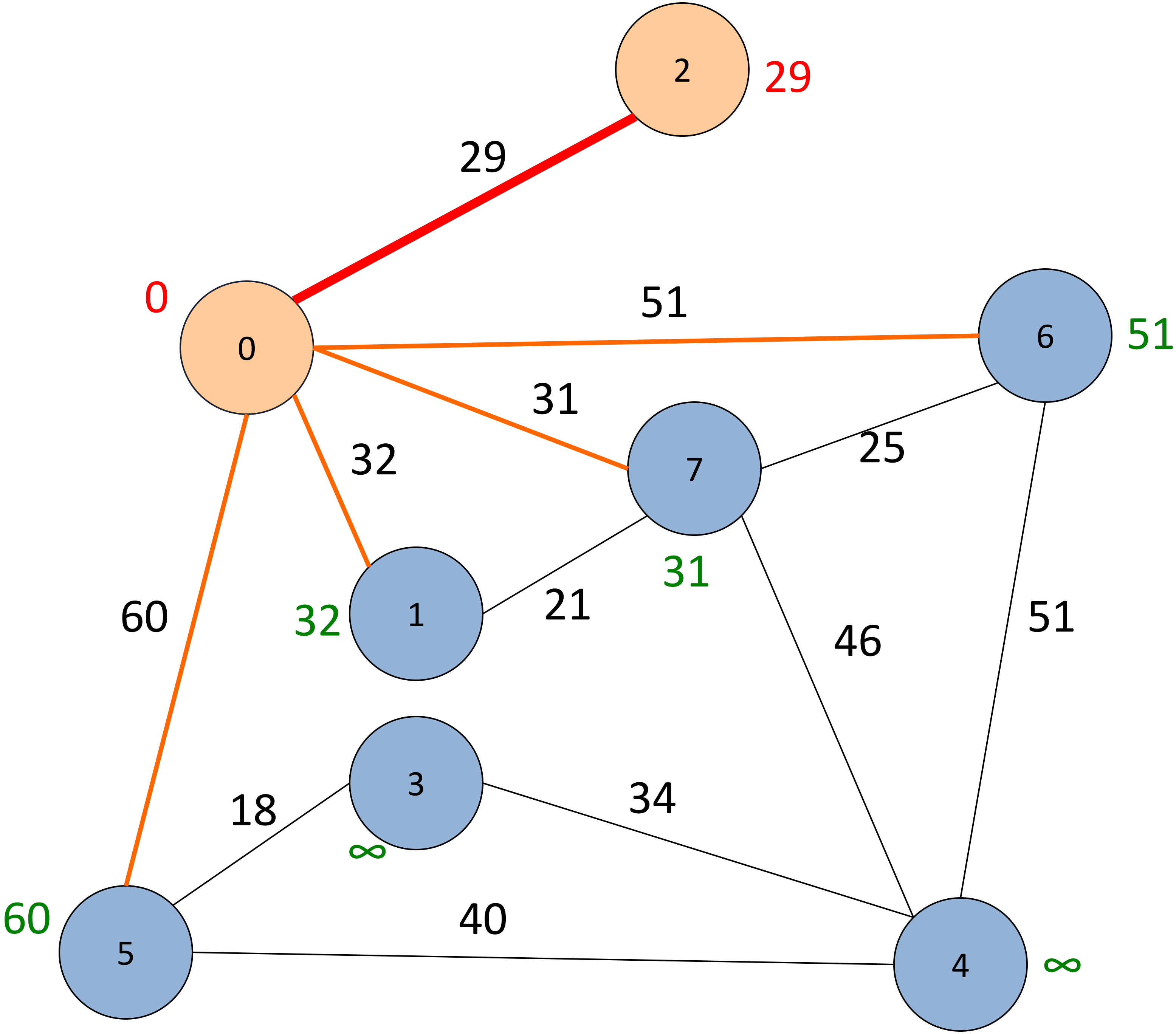
1. Sean $R = \{x\}$, $\overline{R} = V - R$, los nodos incluidos y los que no
2. Sea e la arista de menor costo que cruza de R a $\overline{R}$
3. Sea u el nodo de e que pertenece a $\overline{R}$
4. Agregar e al MST. Eliminar u de $\overline{R}$ y agregarlo a R
5. Si quedan elementos en $\overline{R}$, volver a 2.

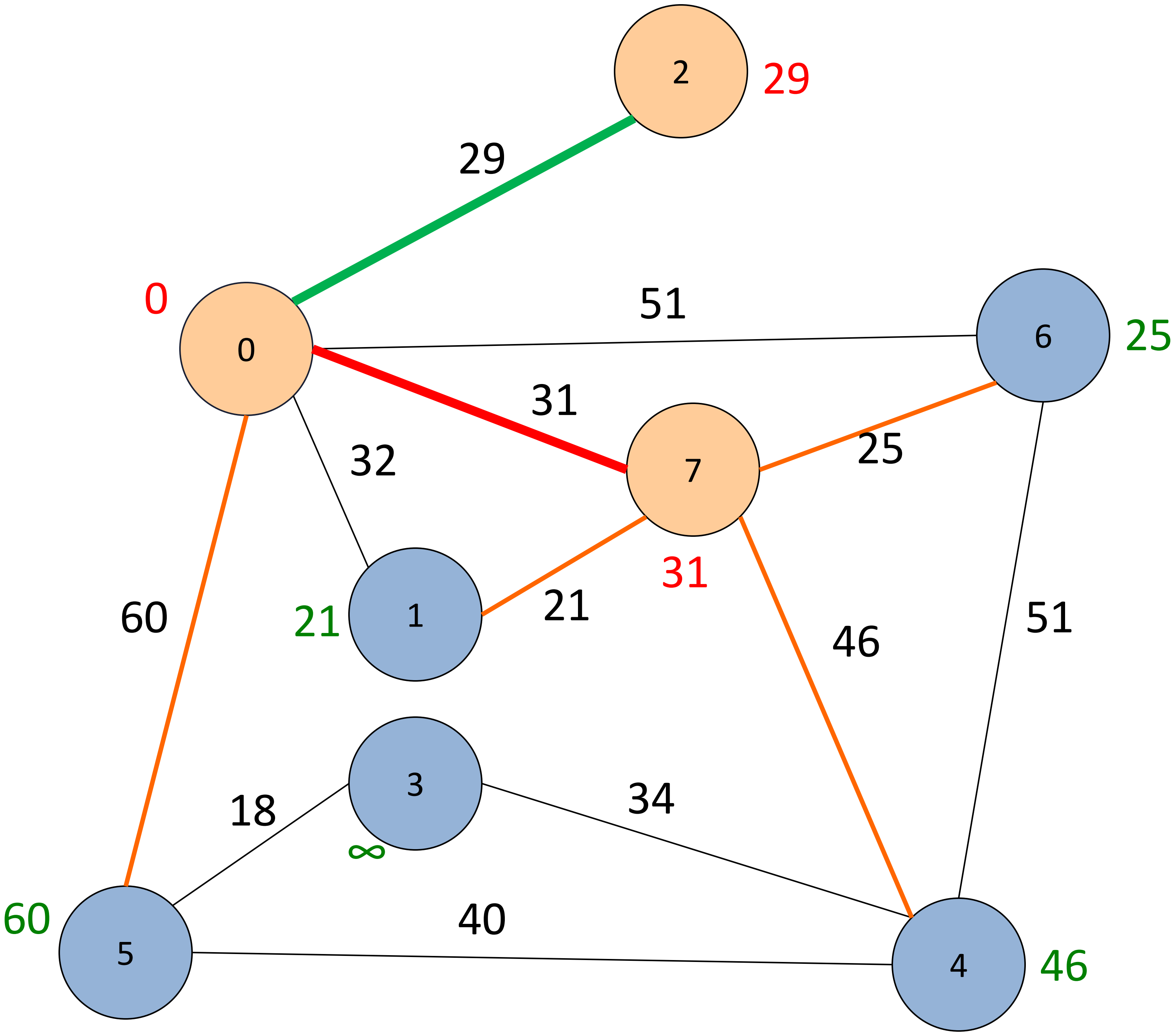

2
29
0
51
6
31
7
25
32
1
21
60
46
51
3
34
18
40
5
4

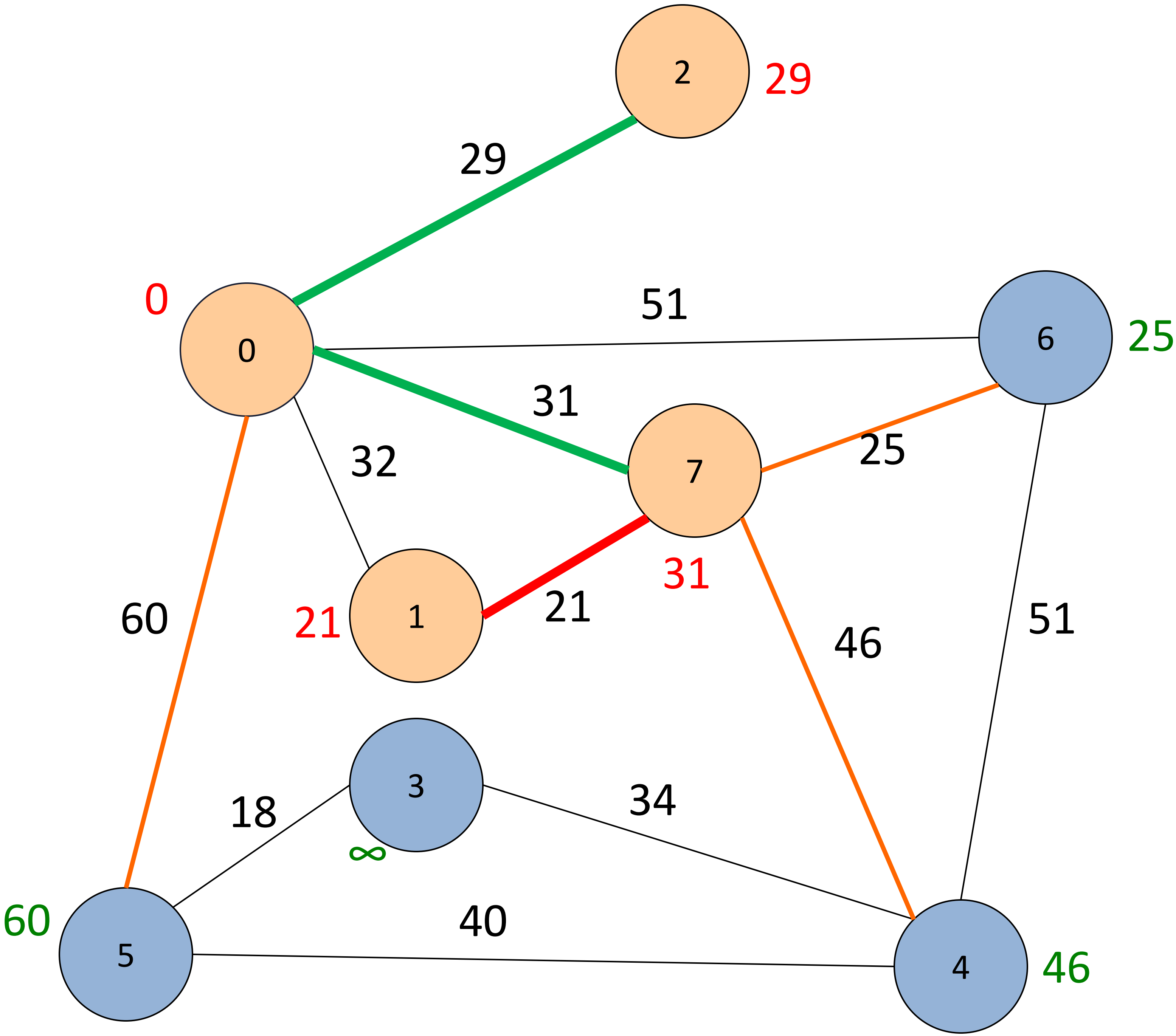

2
∞
29
0
0
51
6
∞
31
7
25
32
∞
60
∞
1
21
46
51
3
18
34
∞
∞
40
5
4
∞



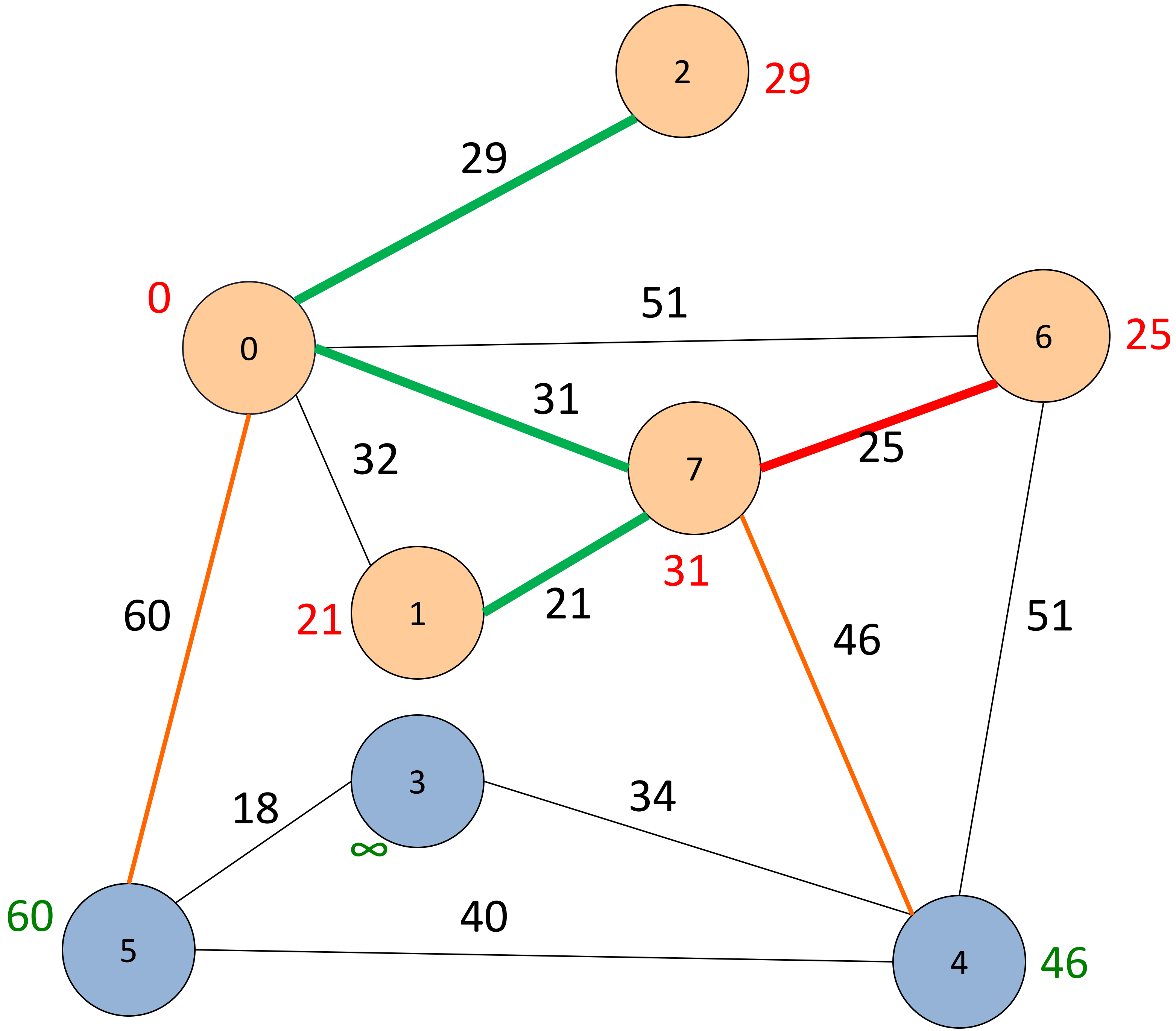
2
29
29
0
0
51
6
51
31
7
25
32
1
21
31
60
32
46
51
3
18
34
∞
60
40
5
4
∞

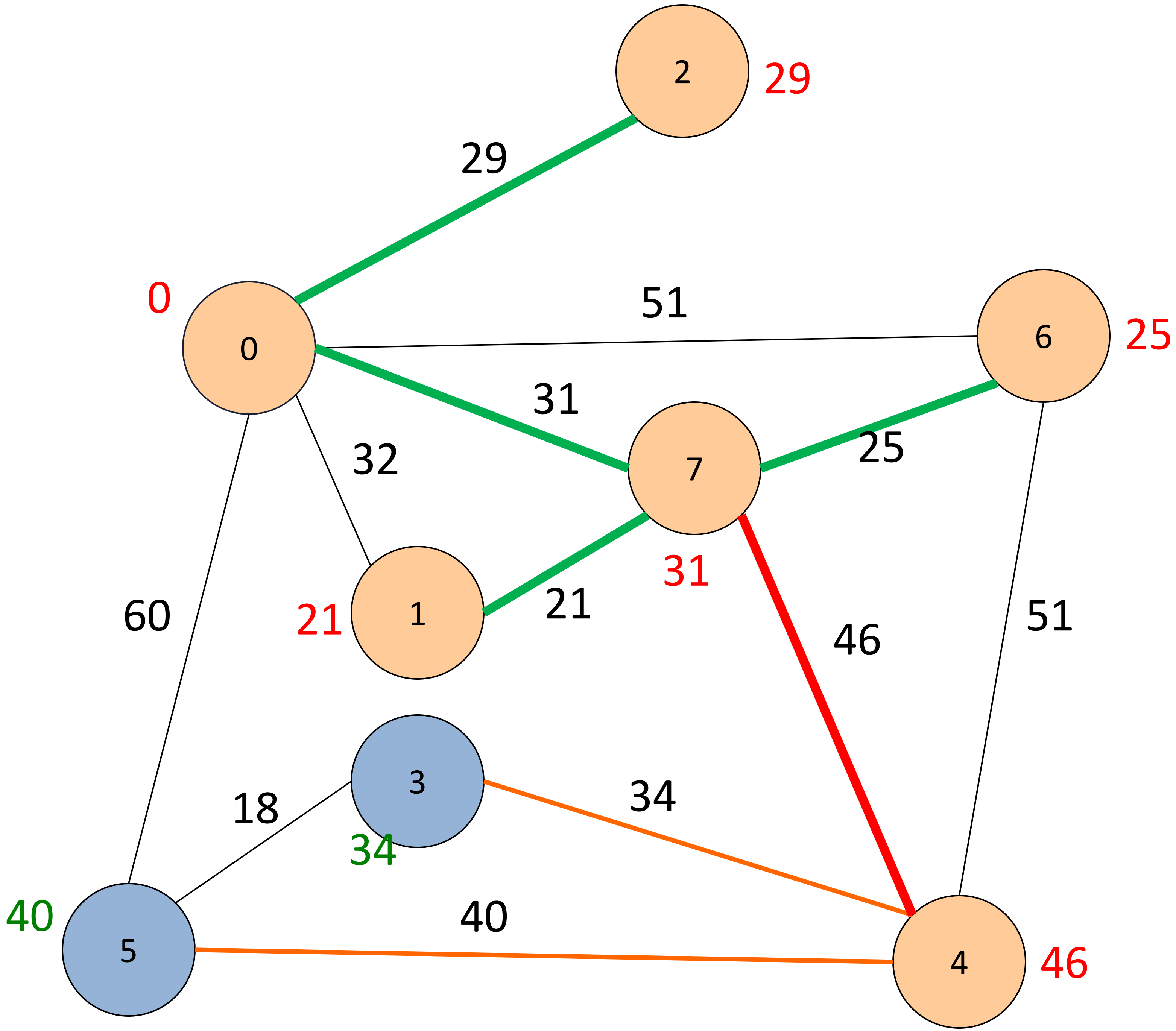

2
29
29
0
0
51
6
51
31
7
25
32
1
32
21
31
60
46
51
3
18
34
∞
60
40
5
4
∞

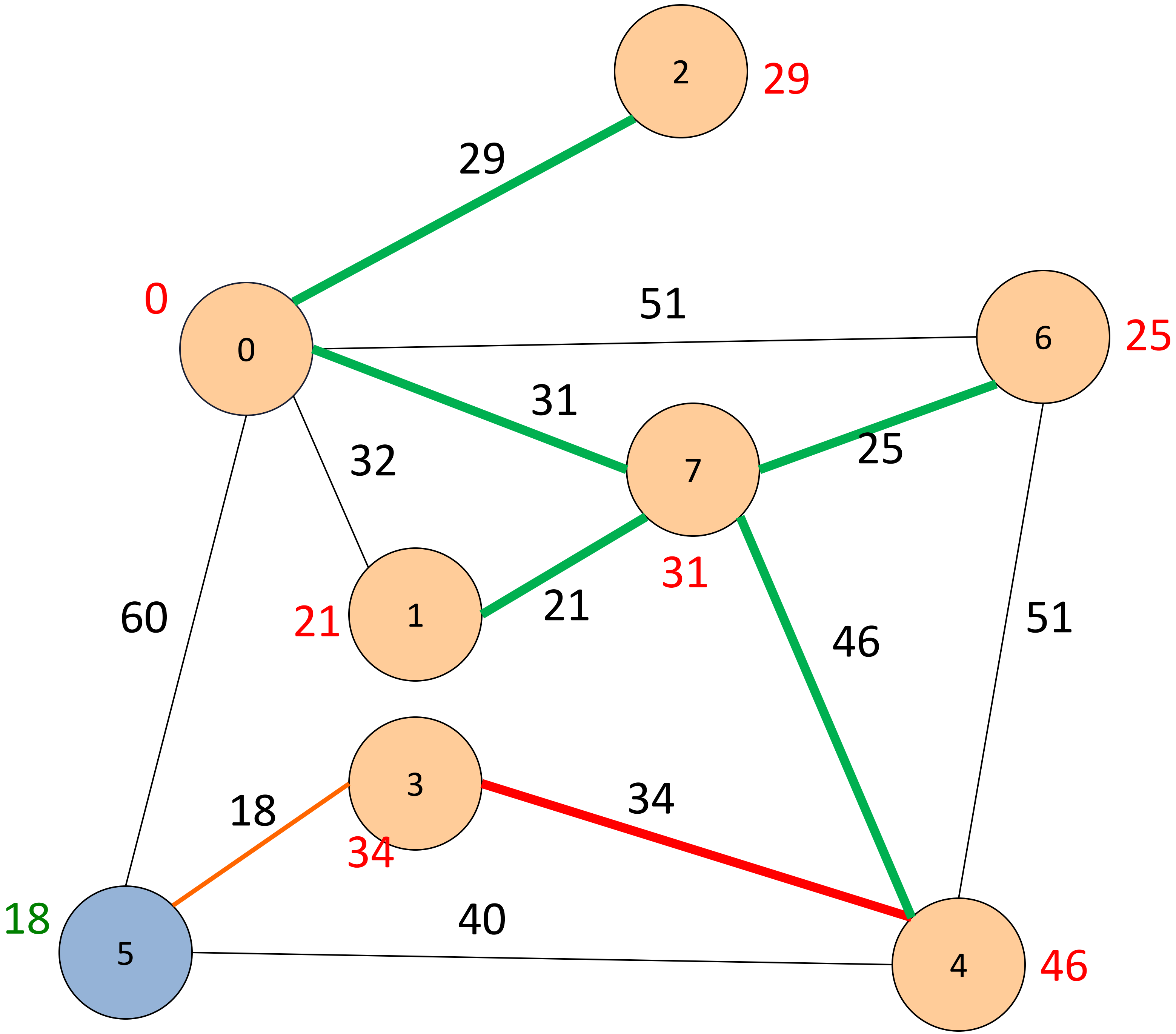

2
29
29
0
51
6
25
0
31
7
25
32
31
60
21
1
21
46
51
3
18
34
∞
60
40
5
4
46



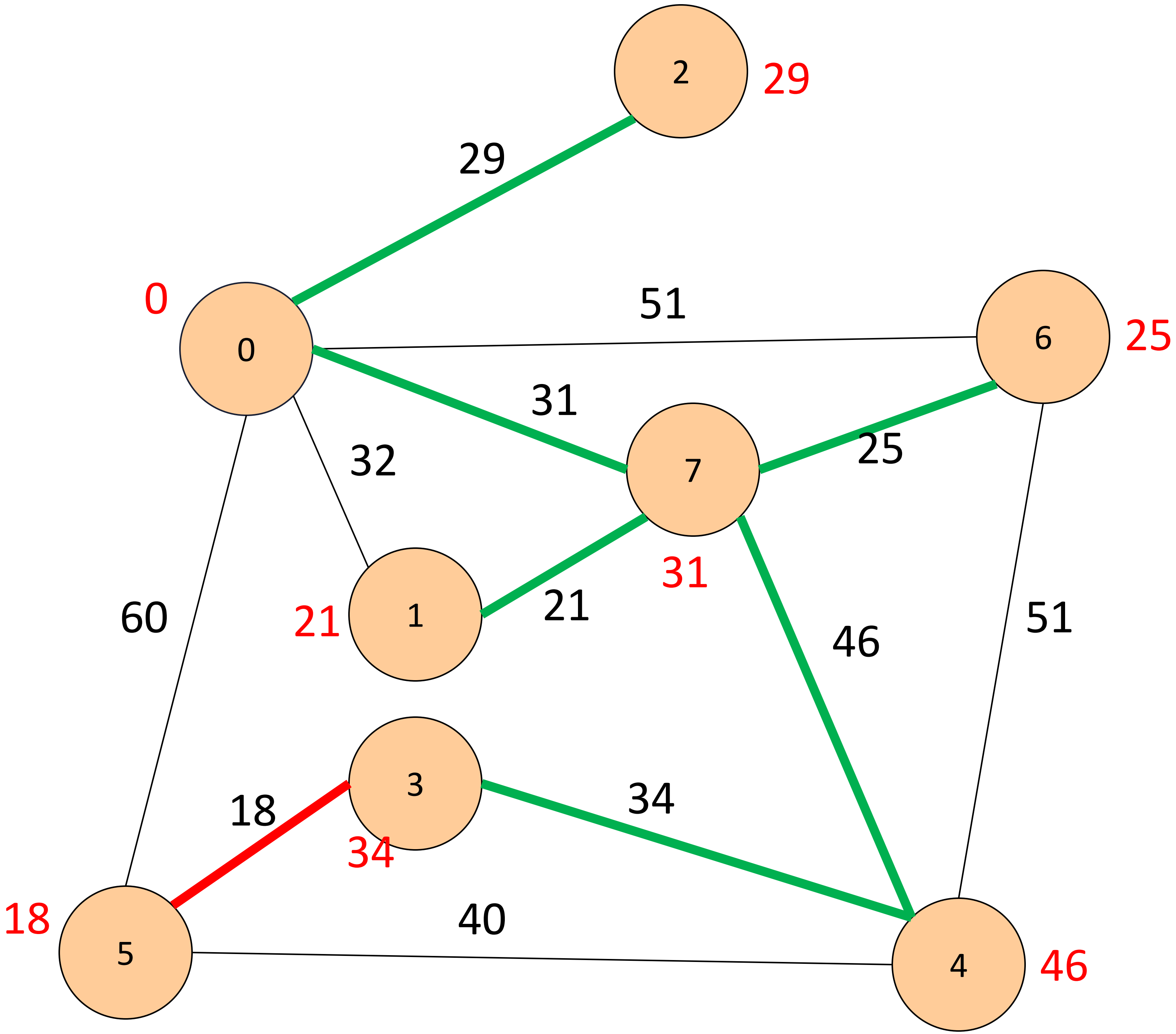
2
29
29
0
0
51
6
25
31
7
32
25
31
1
21
21
60
46
51
3
18
34
∞
60
5
40
4
46

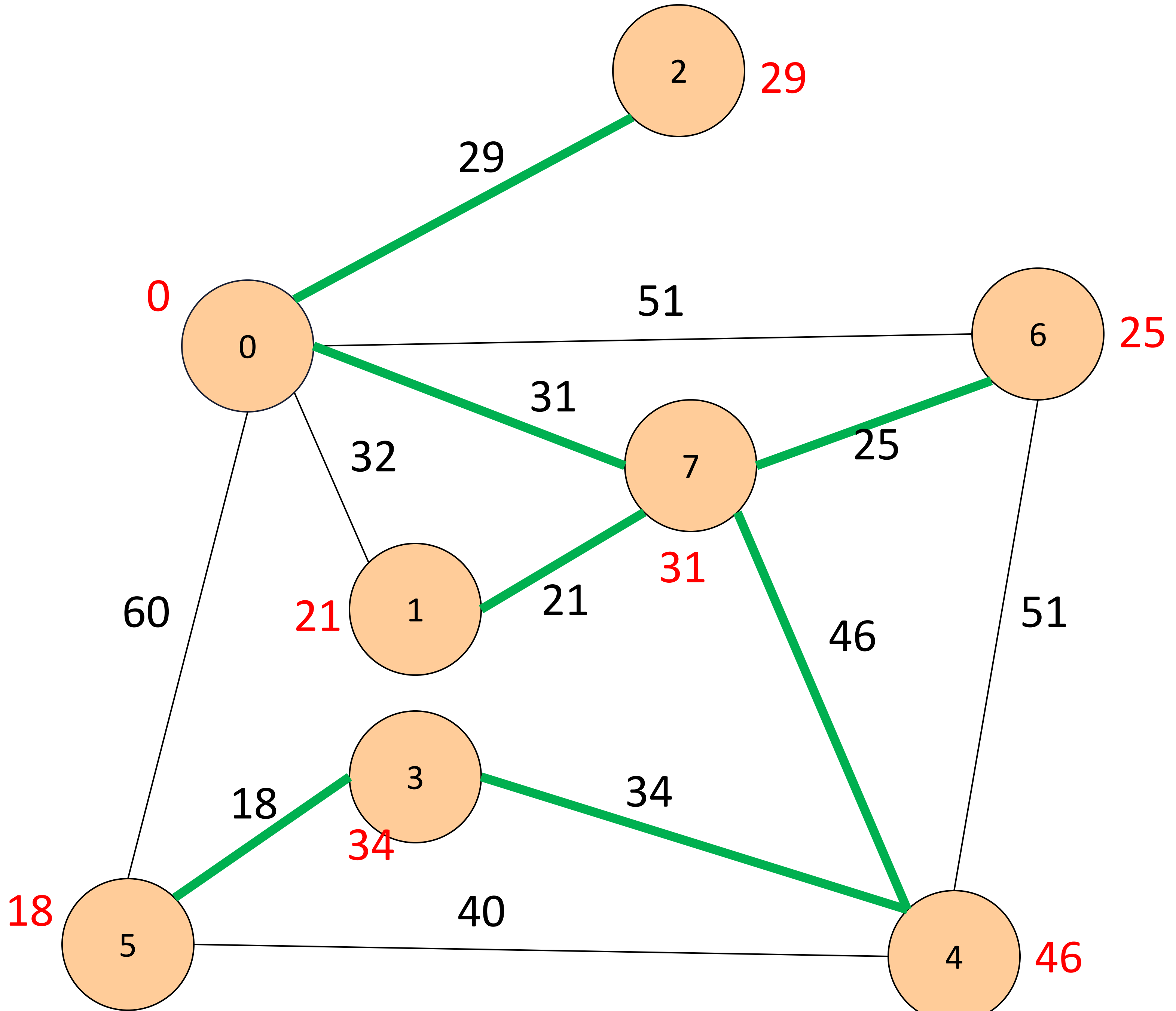

2
29
29
0
0
51
6
25
31
7
25
32
31
60
21
1
21
46
51
3
18
34
∞
60
40
5
4
46


2
29
29
0
0
51
6
25
31
7
25
32
31
1
21
21
60
46
51
3
18
34
34
40
5
40
4
46



2
29
29
0
0
51
6
25
31
32
7
25
31
21
1
21
60
46
51
3
18
34
34
18
40
5
4
46


2
29
29
0
0
51
6
25
31
7
32
25
31
21
1
21
60
46
51
3
18
34
34
18
40
5
4
46


0
1
2
3
4
5
6
7
29
51
31
32
25
60
21
46
51
18
34
40
0
29
25
31
21
34
18
46

$prim(G(V,E),x)$:

$T \leftarrow \emptyset$, $H \leftarrow$ una **cola de prioridades** únicamente con x

$x.key \leftarrow 0$, $x.parent \leftarrow \emptyset$,

$while\ H \neq \emptyset$:

$u \leftarrow$ extraer el vértice de H con menor clave, y pintarlo

$if\ u.parent \neq \emptyset$, agregar la arista $(u.parent,\ u)$ a T

$foreach$ vecino no pintado v de u:

$if\ v \notin H$, insertar v en H

$if\ w(u,v) < v.key$:

$v.key \leftarrow w(u,v)$, $v.parent \leftarrow u$

$return\ T$

Corrección

¿Cómo demostramos que Prim es correcto?

¿Cuál es su complejidad?

Actividades que usan un mismo recurso

Sea $S = \{1, 2, ..., n\}$ un conjunto de n actividades que deben usar un mismo recurso para poder ejecutarse; el recurso puede ser usado por sólo una actividad a la vez:

> cada actividad i tiene una hora de inicio s_i y una hora de término f_i , $s_i \leq f_i$, y, si se ejecuta, transcurre durante el intervalo de tiempo $[s_i , f_i)$
>
> las actividades i y j son **compatibles** si $[s_i, f_i)$ y $[s_j, f_j)$ no se traslapan, es decir, si $s_i \geq f_j$ o $s_j \geq f_i$

El problema consiste en seleccionar un *subconjunto de tamaño máximo de actividades mutuamente compatibles*

Primero, demostraremos que hay una solución óptima que comienza con la elección codiciosa de la actividad 1 (la que termina más temprano):

sea $A \subseteq S$ una solución óptima

ordenemos las actividades en A por hora de término

sea k la primera actividad en A

si $k = 1$, entonces A comienza con una elección codiciosa (y queda demostrado)

...

…

si $k \neq 1$, probamos que hay otra solución óptima B que sí empieza con la actividad 1:

sea $B = A - \{k\} \cup \{1\}$

… entonces como $f_1 \leq f_k$, las actividades en B son compatibles

… y como B tiene el mismo número de actividades que A, B también es una solución óptima (pero que incluye a la actividad 1)

Elegida la actividad 1, el problema se reduce a encontrar una solución óptima al mismo problema,

… pero sobre las actividades en *S* que son compatibles con la actividad 1

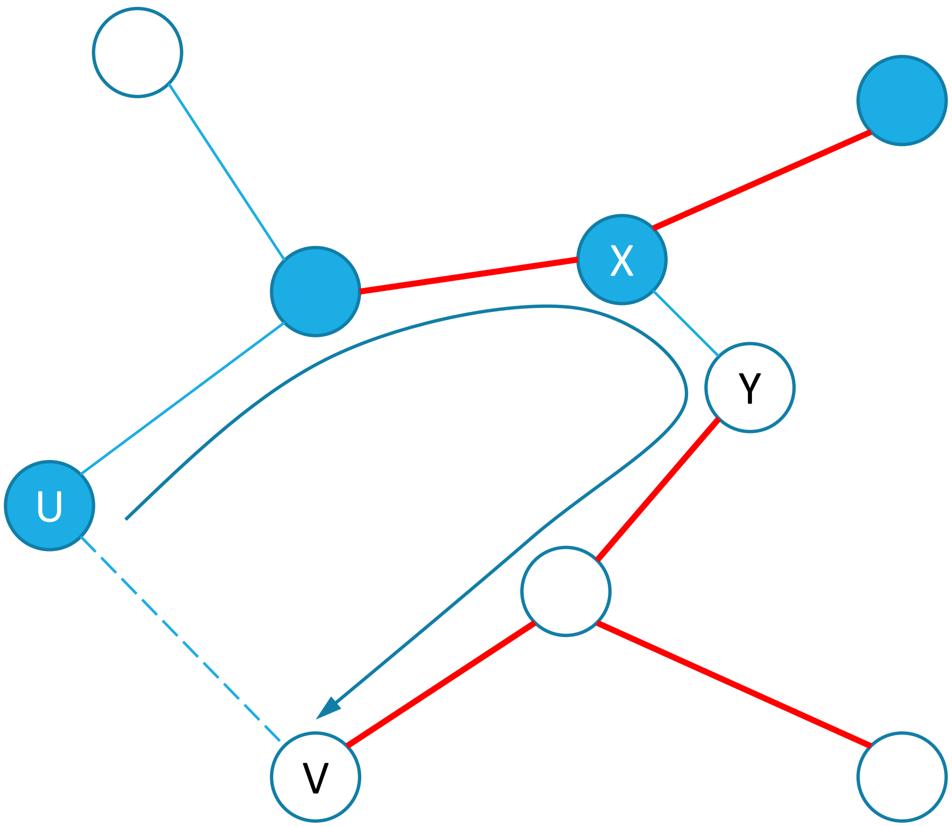
Demostraremos por contradicción que si *A* es una solución óptima a *S*,

… entonces $A' = A - \{1\}$ es una solución óptima a $S' = \{ i \in S : s_i \geq f_1 \}$:

si hubiera una solución *B′* a *S′* con más actividades que *A′*,

… entonces agregando la actividad 1 a *B′* daría una solución *B* a *S* con más actividades que *A*,

… contradiciendo que *A* es óptima


X
Y
U
V

Optimalidad *codiciosa*

Los algoritmos ***greedy*** son muy veloces

Pero no siempre sirven para encontrar el **óptimo**

¿Qué debe cumplirse en un problema para esto?