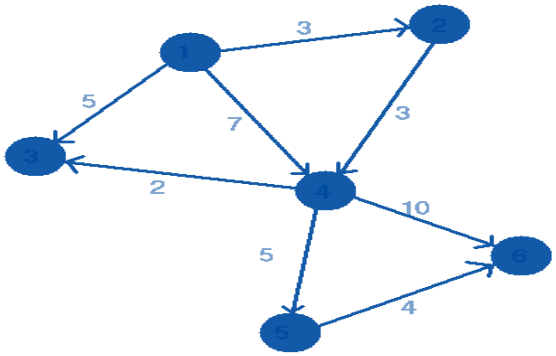


Algorytm Dijkstry oraz algorytm A*

dr hab. inż. Jerzy Balicki, prof. nadzw.

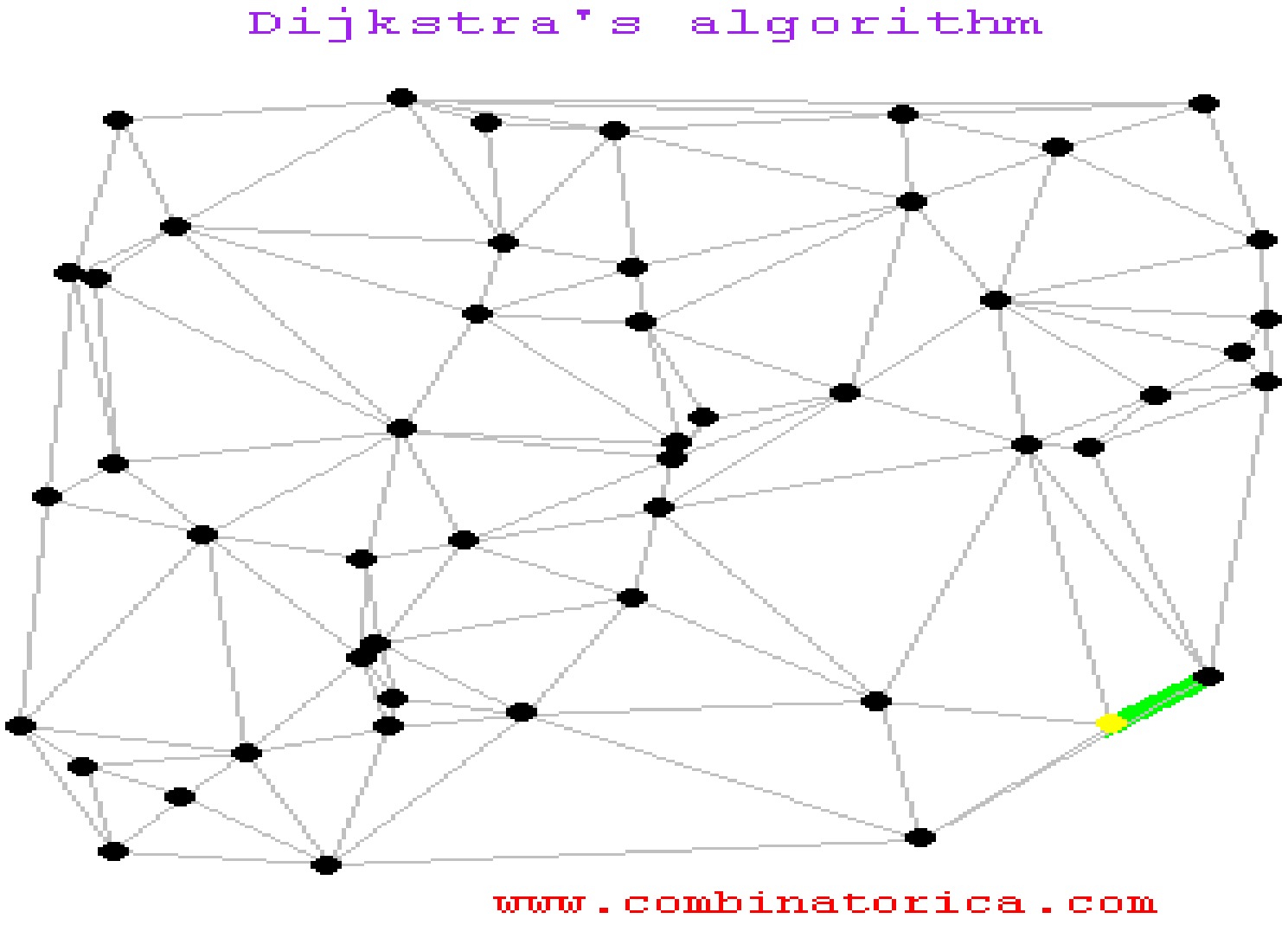
Edsger Wybe Dijkstra

- 1930-2002
- W 1959 r. opracował algorytm wyznaczania najkrótszych ścieżek w grafie (znany jako algorytm Dijkstry)
- w 1972 otrzymał Nagrodę Turinga za wkład w języki programowania

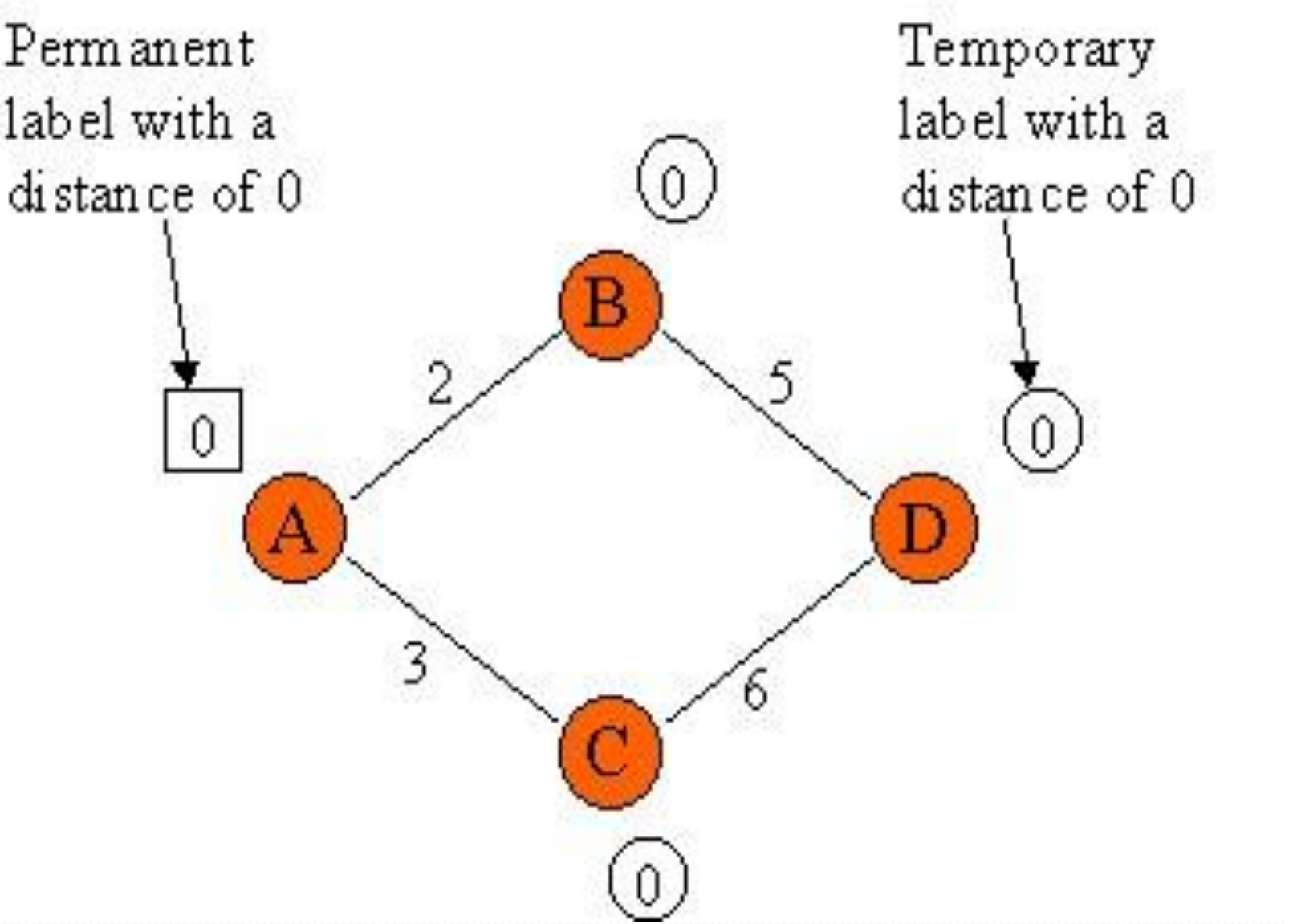
Algorytm Dijkstry

- Algorytm przeszukiwania grafu
- Wyznaczyć najkrótszą ścieżkę w grafie ważonym
- Wagi określone na łukach nie są ujemne
- Dla każdego wierzchołka źródłowego wyznacza najniższy koszt (najkrótszą ścieżkę) do wszystkich wierzchołków w digrafie (lub tylko do pojedynczego wierzchołka)
- W sieciach komputerowych w protokole TCP/IP

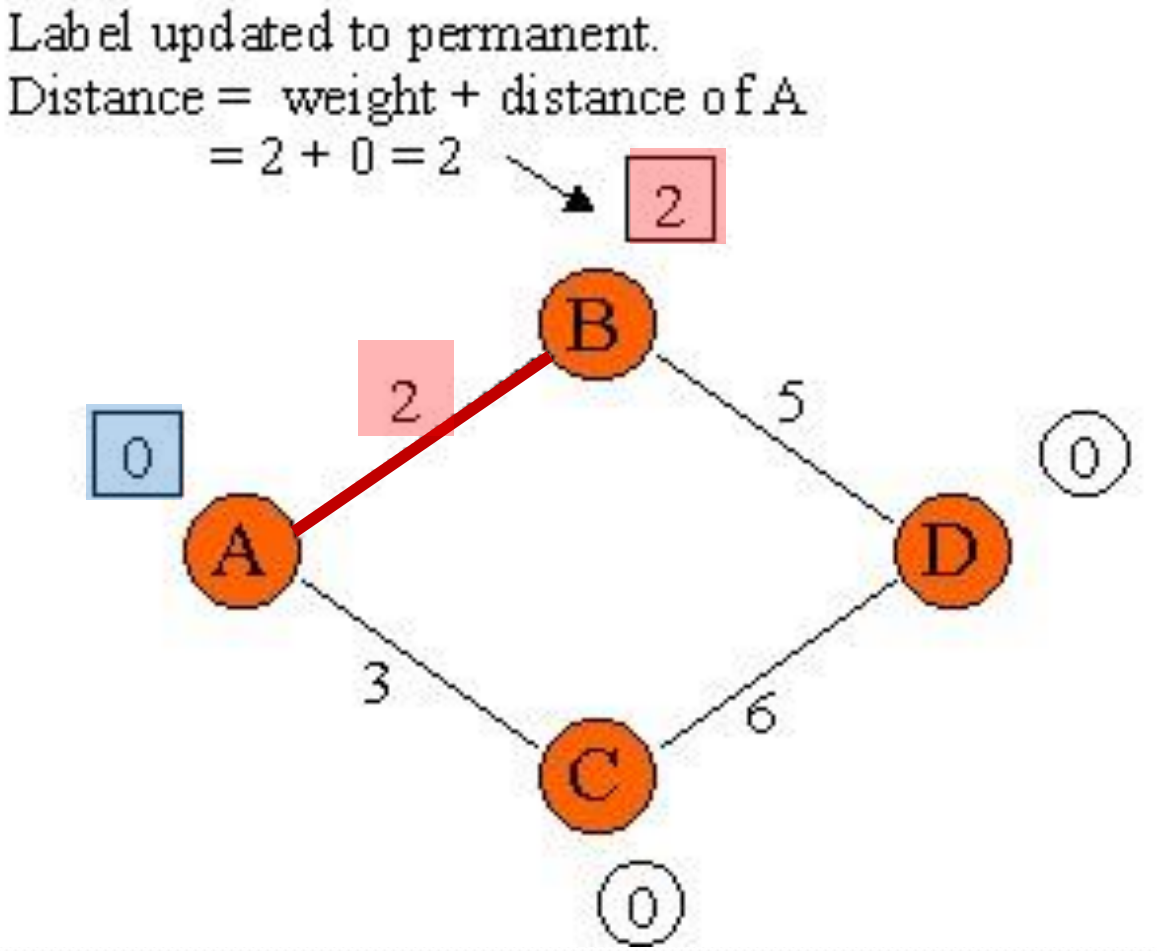
Przykład



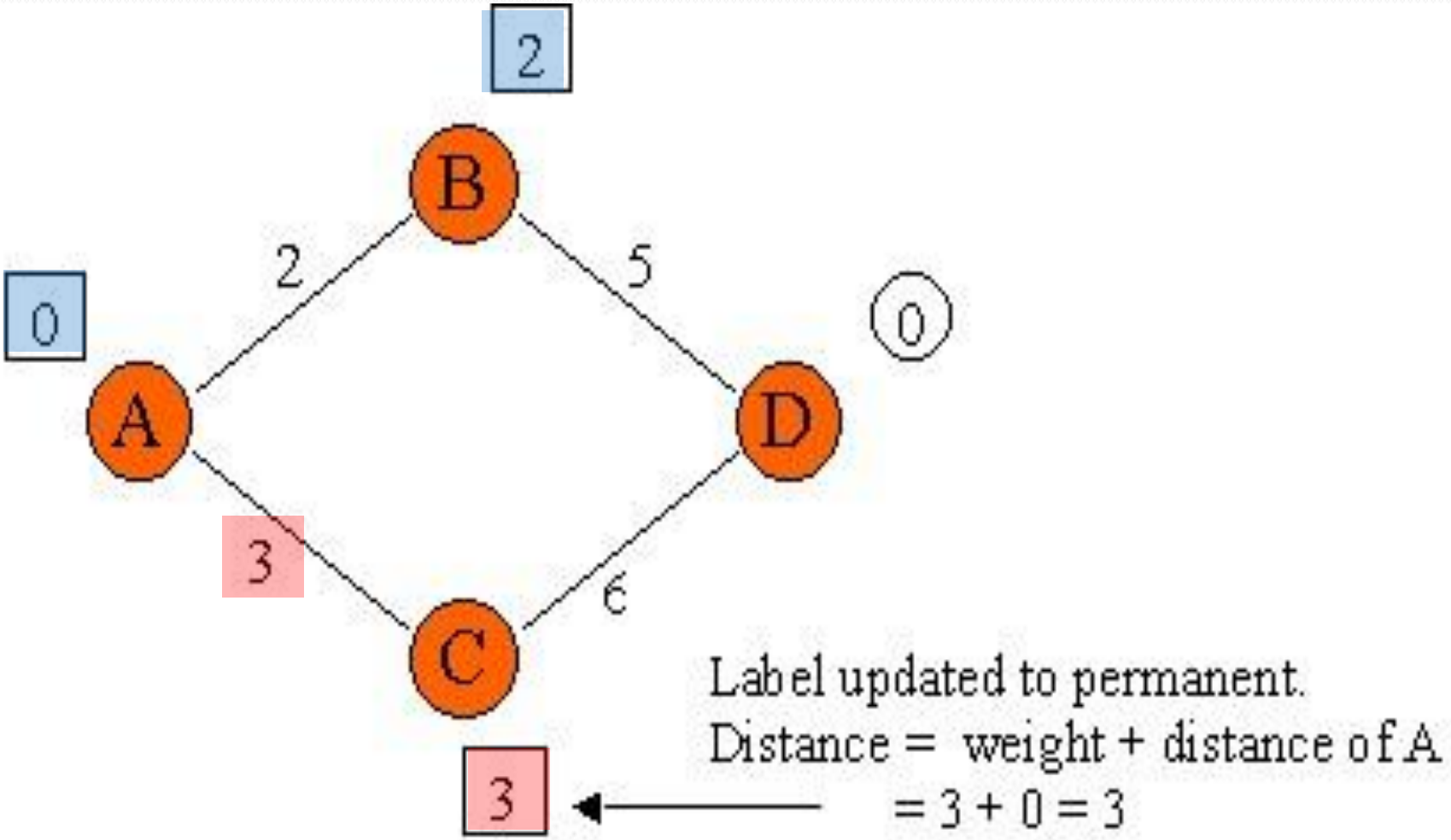
Inicjalizacja wartości zerowych etykiet dla wierzchołków



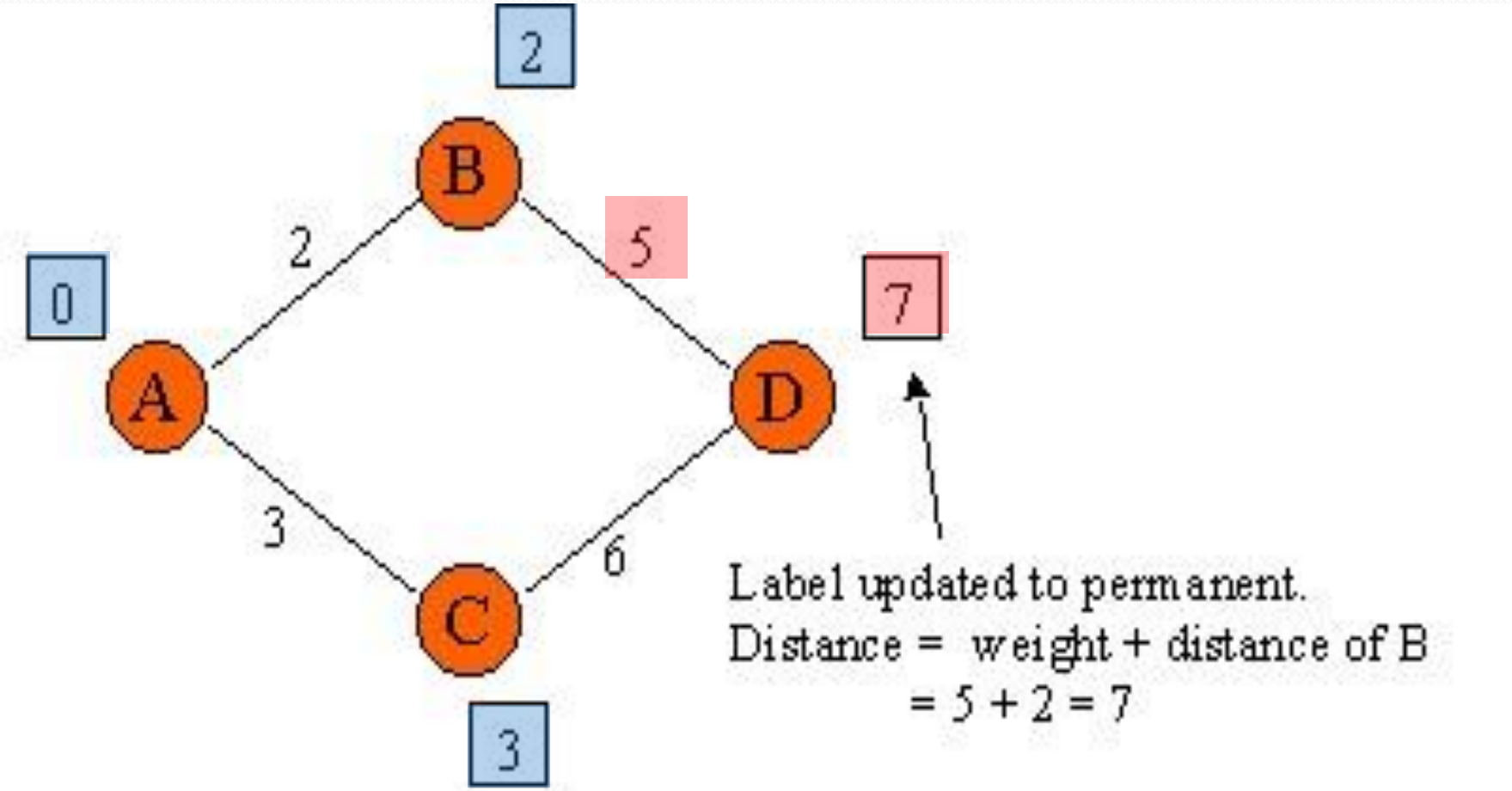
Ocechowanie wierzchołka przyległego połączonego za pomocą najtańszego łuku



Ocechowanie „najtańszego” wierzchołka przyległego do *A* lub *B*



Ocechowanie pozostałych wierzchołków



Algorytm Dijkstry

Niech s - wierzchołek źródłowy, $w(i,j)$ - waga krawędzi (i,j),
d - tablica odległości od źródła s do wierzchołków grafu.
Na początku $d[s] = 0$, zaś $d[v] = \infty$ dla pozostałych węzłów.

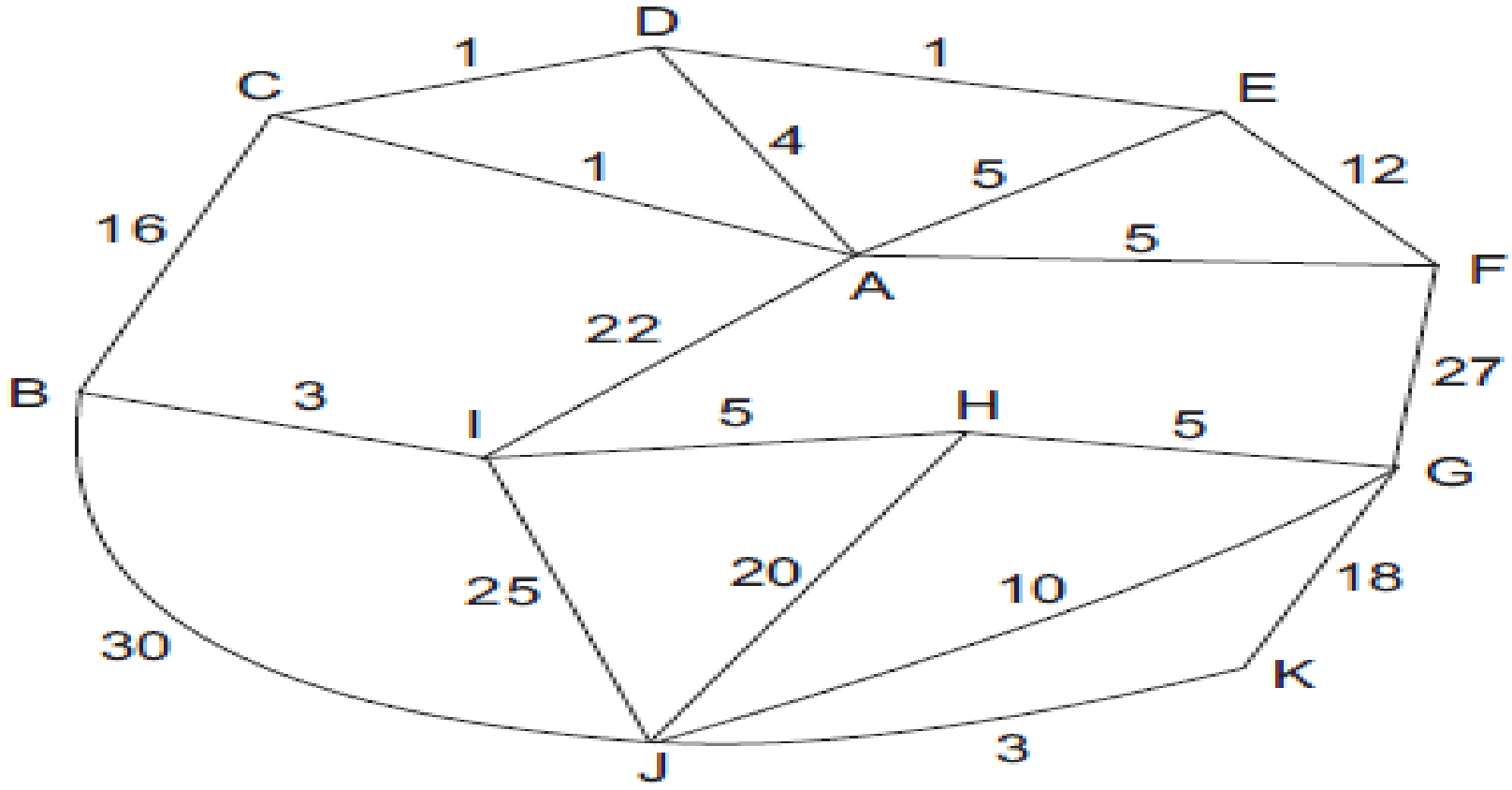
1. Utwórz kolejkę priorytetową Q wierzchołków. Priorytetem jest aktualnie wyliczona odległość od s.
2. Dopóki kolejka nie jest pusta:
   Usuń z kolejki wierzchołek u o najniższym priorytecie (najbliższy źródła, który nie został jeszcze **odwiedzony**)
   Dla każdego sąsiada v węzła u dokonaj *relaksacji*:
   jeśli $d[u] + w(u,v) < d[v]$ (poprzez u da się dojść do v szybciej niż dotychczasową ścieżką), to $d[v] := d[u] + w(u,v)$.

Na końcu tablica d zawiera najkrótsze odległości do wszystkich wierzchołków.
Dodatkowo, możemy w tablicy *poprzednik* przechowywać dla każdego wierzchołka numer jego bezpośredniego poprzednika na najkrótszej ścieżce, co pozwoli na odtworzenie pełnej

Algorytm Dijkstry

Dodatkowo, możemy w tablicy *poprzednik* przechowywać dla każdego wierzchołka numer jego bezpośredniego poprzednika na najkrótszej ścieżce, co pozwoli na odtworzenie pełnej ścieżki od źródła do każdego wierzchołka – przy każdej relaksacji w ostatnim punkcie, *u* staje się poprzednikiem *v*.

Algorytm Dijkstry



Otwarte											
Zamknięte											

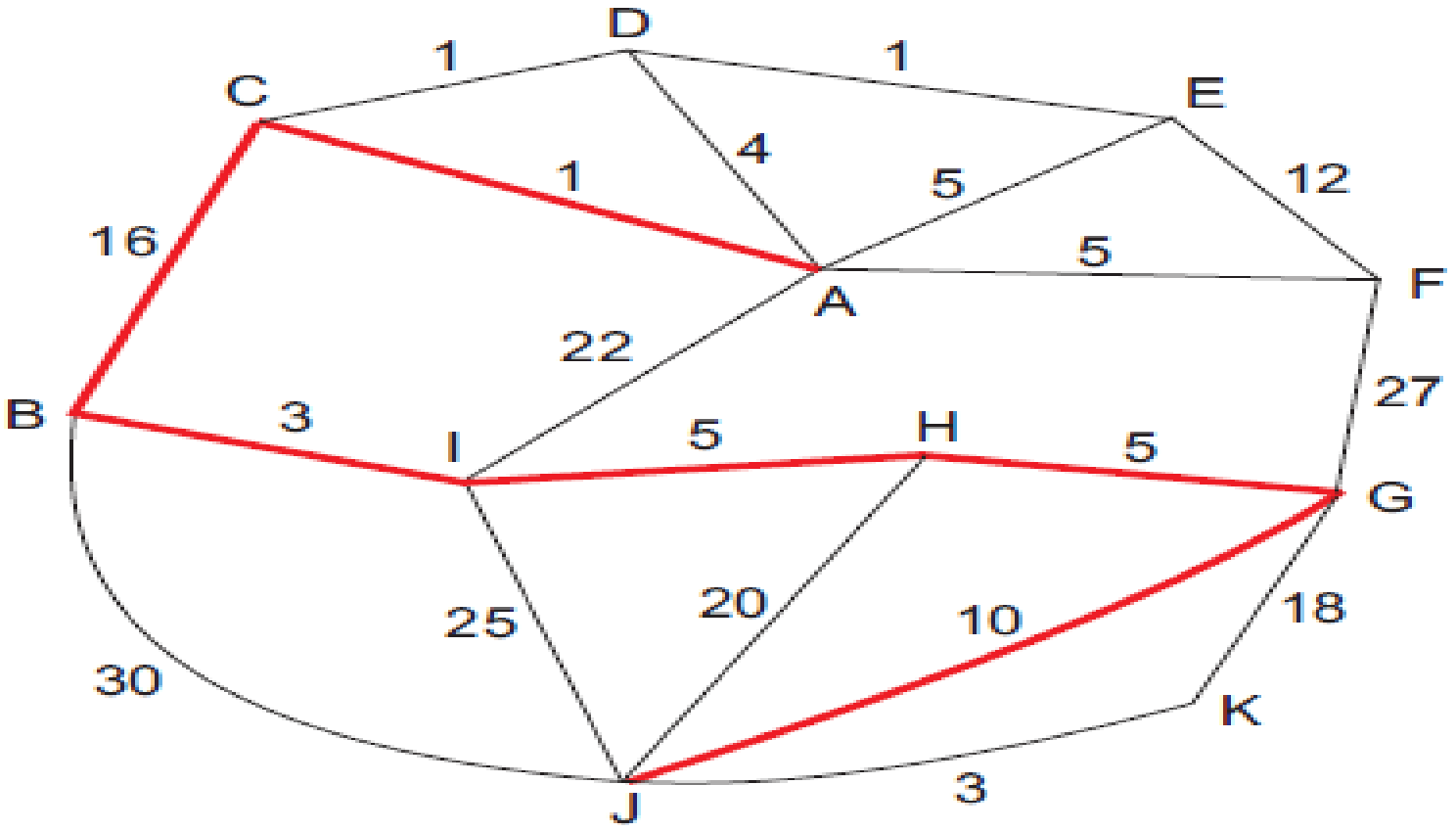
Algorytm Dijkstry

Otwarte	0 A										
Zamknięte											

Otwarte	0 ~~A~~										
Zamknięte	A										

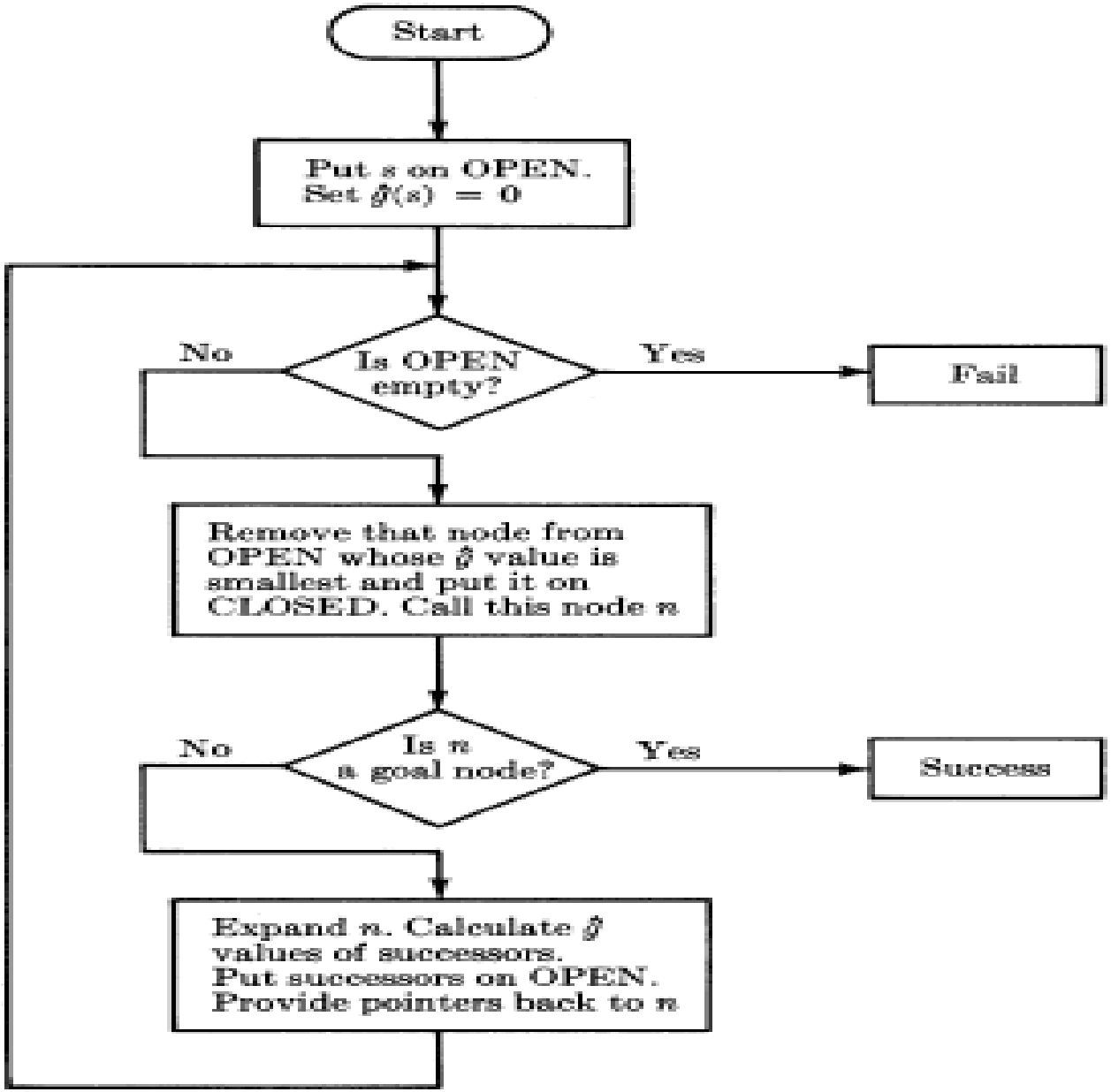
Otwarte	0 ~~A~~	1 C	4 D	5 E	5 F	22 I					
Zamknięte	A										

Algorytm Dijkstry



			2	3		20		30	40 45		
Otwarte	0 ~~A~~	1 ~~C~~	4 ~~D~~	5 ~~E~~	5 ~~F~~	22 ~~I~~	17 ~~B~~	32 ~~G~~	47 ~~J~~	25 ~~H~~	48 K
Zamknięte	A	C	D	E	F	B	I	H	G	J	

Algorytm Dijkstry



Złożoność obliczeniowa algorytmu Dijkstry

Niech V - liczba wierzchołków i E – liczba krawędzi grafu.

O rzędzie złożoności decyduje implementacja kolejki priorytetowej.

1. Stosując tablicę, otrzymujemy złożoność $O(V^2)$
2. Stosując implementację kolejki poprzez kopiec, złożoność wynosi $O(E \log V)$
3. Stosując kopiec Fibonacciego, złożoność zmniejsza się do $O(E + V \log V)$

Pierwszy wariant jest optymalny dla grafów gęstych (jeśli $E = \Theta(V^2)$),
drugi jest szybszy dla grafów rzadkich ($E = \Theta(V)$), trzeci jest bardzo rzadko używany ze względu na duży stopień skomplikowania i niewielki w porównaniu z nim zysk czasowy.

Algorytm A*

- **Algorytm *A** jest pewnym uogólnieniem algorytmu Dijkstry, które pozwala przeszukiwać tylko część grafu, jednak wymaga dodatkowej wstępnej informacji (heurystyki) o odległościach wierzchołków.**
- **Algorytm został opisany w 1968 roku przez Petera Harta, Nilsa Nilssona oraz Bertrama Raphaela.**

Algorytm A*

1. Od *wierzchołka początkowego* tworzy się *ścieżkę*, za każdym razem wybierając wierzchołek x z dostępnych w danym kroku *niezbadanych wierzchołków* tak, by minimalizować funkcję $f(x)$:

$$f(x) = g(x) + h(x)$$

gdzie:

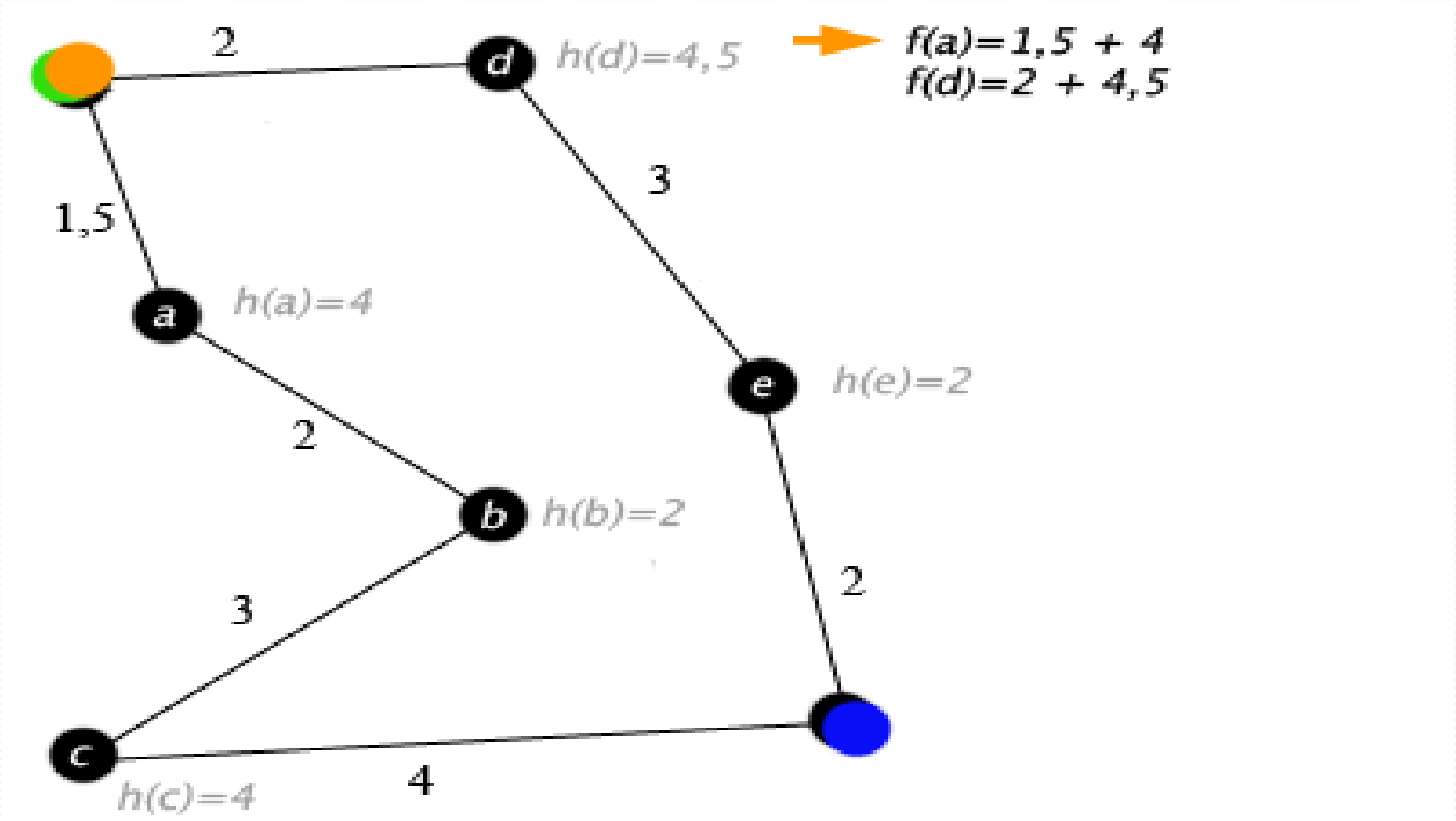
$g(x)$ – koszt drogi między *wierzchołkiem początkowym* a x (suma wag krawędzi, które należą już do ścieżki plus waga krawędzi łączącej *aktualny węzeł* z x)

$h(x)$ – przewidywana przez *heurystykę* droga od x do *wierzchołka docelowego*.

2. W każdym kroku algorytm dołącza do ścieżki wierzchołek o najniższym współczynniku f.
3. STOP w momencie natrafienia na wierzchołek będący *wierzchołkiem docelowym*.

Algorytm A*

Niech wierzchołkami będą miasta, wagami krawędzi - odległości drogowe, a **heurystyka *h(x)* jest odległością w linii prostej.**



References

- E. W. Dijkstra: *A note on two problems in connexion with graphs*. In *Numerische Mathematik*, 1 (1959), S. 269–271.
- http://students.ceid.upatras.gr/~papagel/project/kef5_7_1.htm
- http://www.animal.ahrgr.de/showAnimationDetails.php3?lang=en&anim=16
- http://www.cs.sunysb.edu/~skiena/combinatorica/animations/dijkstra.html
- http://www.ibiblio.org/links/devmodules/graph_networking/xhtml/page13.xml
- Survivable networks: algorithms for diverse routing By Ramesh Bhandari Edition: illustrated Published by Springer, 1999 ISBN page 22