

1 Algorytm Prima

```
Start
Wczytaj graf nieskierowany graph w postaci macierzowej
vertex_count ← ilość wierzchołków graph
weights ← tablica długości vertex_count
history ← tablica długości vertex_count
Dla  $i \leftarrow 0$  do vertex_count wykonuj
| weights[i] ←  $\infty$ 
| history[i] ← brak
koniec
pending_vertices ← set pojemności vertex_count
Dla  $i \leftarrow 0$  do vertex_count wykonuj
| Dodaj  $i$  do pending_vertices
koniec
weights[0] ← 0
Gdy długość pending_vertices > 0 wykonuj
| vertex ← element z pending_vertices o najmniejszej wartości weights[vertex]
| Usuń wartość vertex z pending_vertices
| Dla neighbor ← 0 do vertex_count wykonuj
| | distance ← graph[vertex][neighbor]
| | Jeżeli distance  $\leq 0$  lub neighbor nie występuje w pending_vertices to
| | | Przejdź do kolejnej iteracji pętli
| | koniec
| | Jeżeli distance < weights[neighbor] to
| | | weights[neighbor] ← distance
| | | history[neighbor] ← vertex
| | koniec
| koniec
koniec
edges ← tablica rozmiaru vertex_count  $\times$  vertex_count wypełniona 0
Dla vertex ← 0 do vertex_count wykonuj
| weight ← weights[vertex]
| Jeżeli weight =  $\infty$  to
| | Przejdź do kolejnej iteracji pętli
| koniec
| neighbor ← history[vertex]
| edges[vertex][neighbor] ← weight
| edges[neighbor][vertex] ← weight
koniec
Wypisz edges
Stop
```