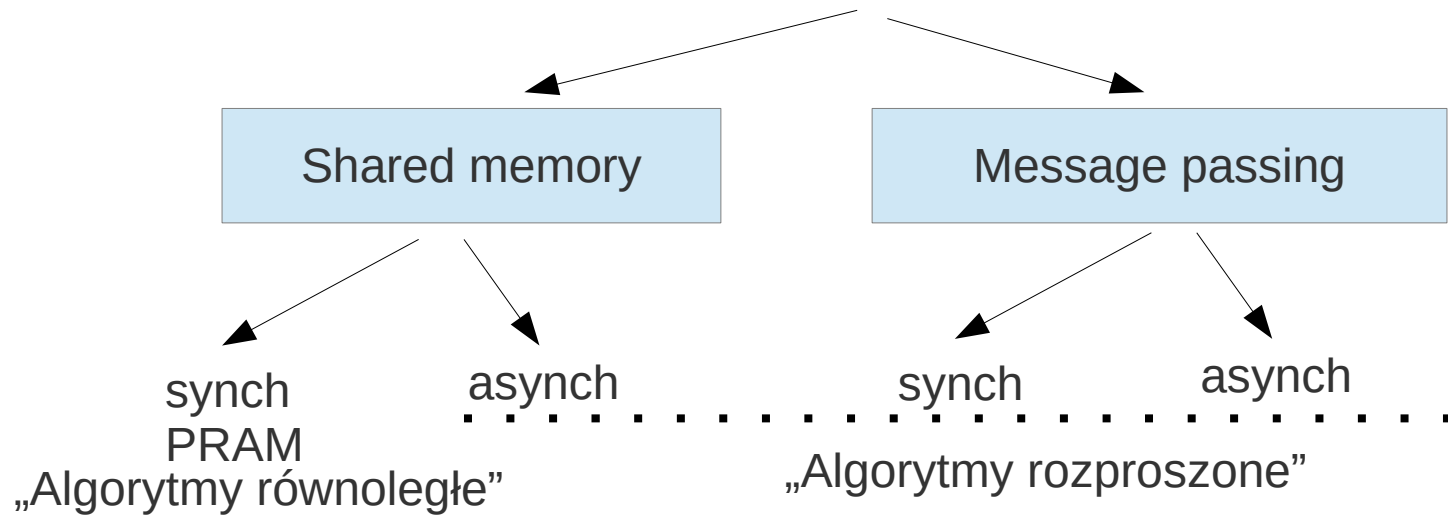


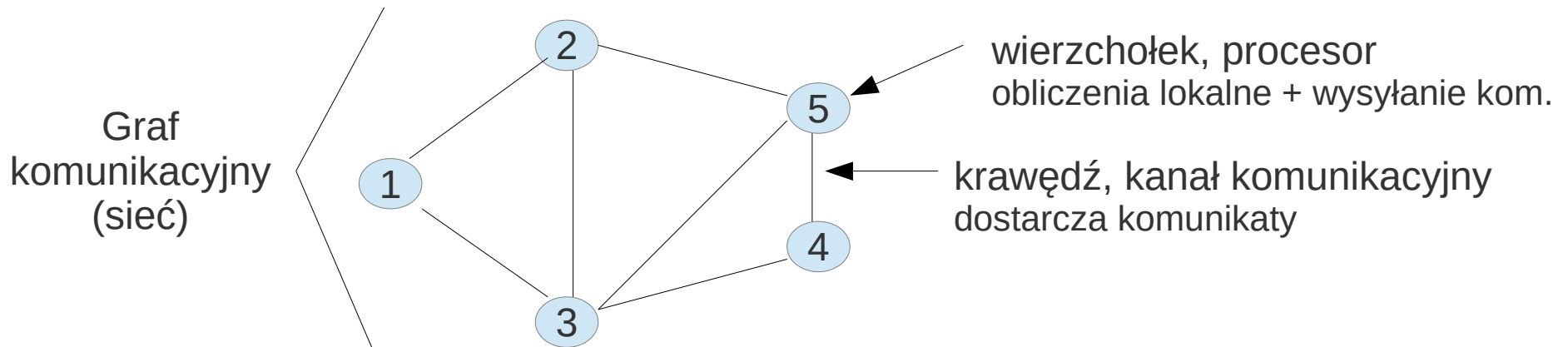
# ALR

Algorytmy rozproszone  
*(algorytmy, nie systemy!!)*

# Modelujemy systemy n-procesorowe...



Model „Message passing” - nieformalna definicja :



**Co jest celem alg rozproszonego?** Odp: obliczenie „czegoś” w grafie kom

# Model message-passing – synch i asynch

W modelu synch mamy **rundy**...

w każdej rundzie każdy wierzchołek:

1. wykonuje obliczenia lokalne („za darmo”)
2. wymienia komunikaty z sąsiadami w grafie kom.

(dokładniej: w  $i+1$  rundzie otrzymuje kom wysłane w  $i$ -tej rundzie przez sąsiadów)

W modelu asynch nie ma rund...

wysłane przez kraw komunikaty po jakimś czasie są dostarczane...

*Warianty modelu message-passing:*

1. graf kom może być: cyklem (ring), drzewem, pełnym grafem, ...
2. stopień synchronizacji: synch i asynch
3. stopień symetrii, czy wierz  $v$  ma  $ID(v)$  czy też są nieodróżnialne?
4. jednorodność (uniform), niejednorodny= wierz muszą znać „ $n$ ” (liczbę wierz)

**Formalna def modelu message-passing, asynch:**

mamy  $n$ -procesorów,  $P_1, \dots, P_n$

**konfiguracja**,  $C = (q_1, \dots, q_n)$ , stan procesorów (zmienne), także zawartość połączeń

**zdarzenie**, 1. obliczeniowe (obl lokalne + wysyłanie kom), 2. dostarczenia kom

**egzekucja**,  $C_0, \Phi_0, C_1, \Phi_1, \dots$  gdzie  $C_i$  to konfiguracja,  $\Phi_i$  to zdarzenie

konfiguracja  $C_{i+1}$  wynika z zastosowania zdarzenia  $\Phi_i$  do konfig  $C_i$

w przypadku alg deterministycznych wystarczy podać:  $C_0, \Phi_0, \Phi_1, \Phi_2, \dots$

# Model message-passing – synch i asynch

Co to znaczy, że dany algorytm **asynch** „A” działa ?

Odp: działa (obl to co trzeba) dla dowolnej dopuszczalnej egzekucji

Egzekucja jest dopuszczalna gdy:

1. każdy wierz ma nieskoczenie wiele zdarzeń obliczeniowych,
2. każdy wysłany komunikat jest dostarczany (kiedy?)

**Formalna definicja modelu message-passing, synch:**

rozpatrujemy egzekucje dopuszczalne, które dają się podzielić na rundy, tj takie, że w każdej rundzie:

1. każdy wierz ma 1 zdarzenie obl,
2. wszystkie wysłane kom są dostarczane (w tej rundzie)

wysłane w i-tej rundzie są dostępne w (i+1)-rundzie



Co to znaczy, że dany algorytm **synch** „A” działa ?

Odp: działa dla każdej egz, która można podzielić na rundy...

Uwaga: alg synch jest asnych !!!

# Złożoność obliczeniowa w modelu mess-pass

Synch:

- czas (liczba rund)
- liczba kom

Asynch:

- liczba kom
- czas (???)

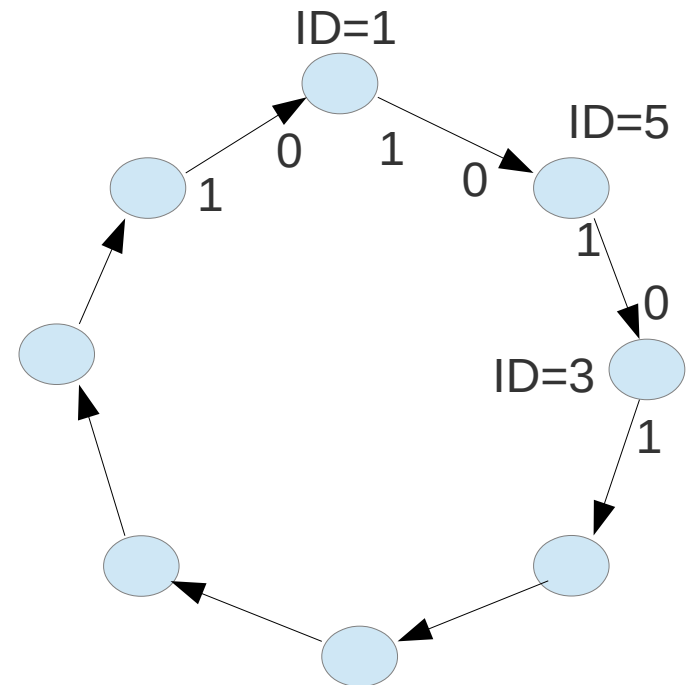
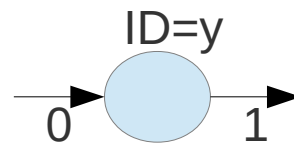
*Problem:* leader election, **LE**, ring (zorient), asynch, są ID, podamy alg, w którym licz kom =  $O(n^2)$

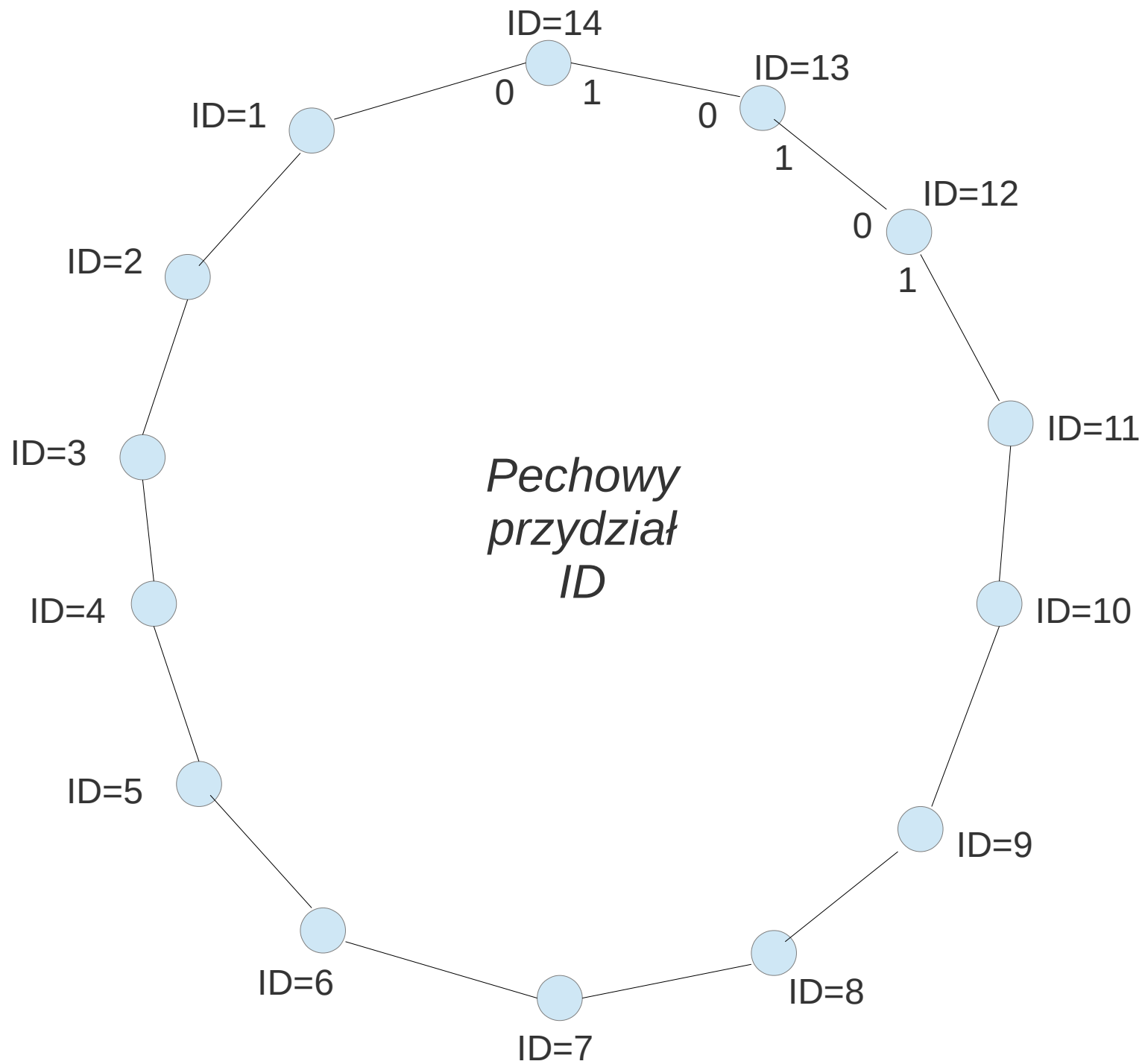
**alg:**

1. każdy wierz wysyła swój ID przez poł 1
2. gdy z poł 0 przyjdzie X, to:
  - jeśli  $X > ID$  to wysyłamy X przez poł 1
  - jeśli  $X == ID$  to ogłaszamy się liderem i zawiadamiamy pozostałe wierz, że nie są liderami
  - jeśli  $X < ID$  to ... nic nie robimy

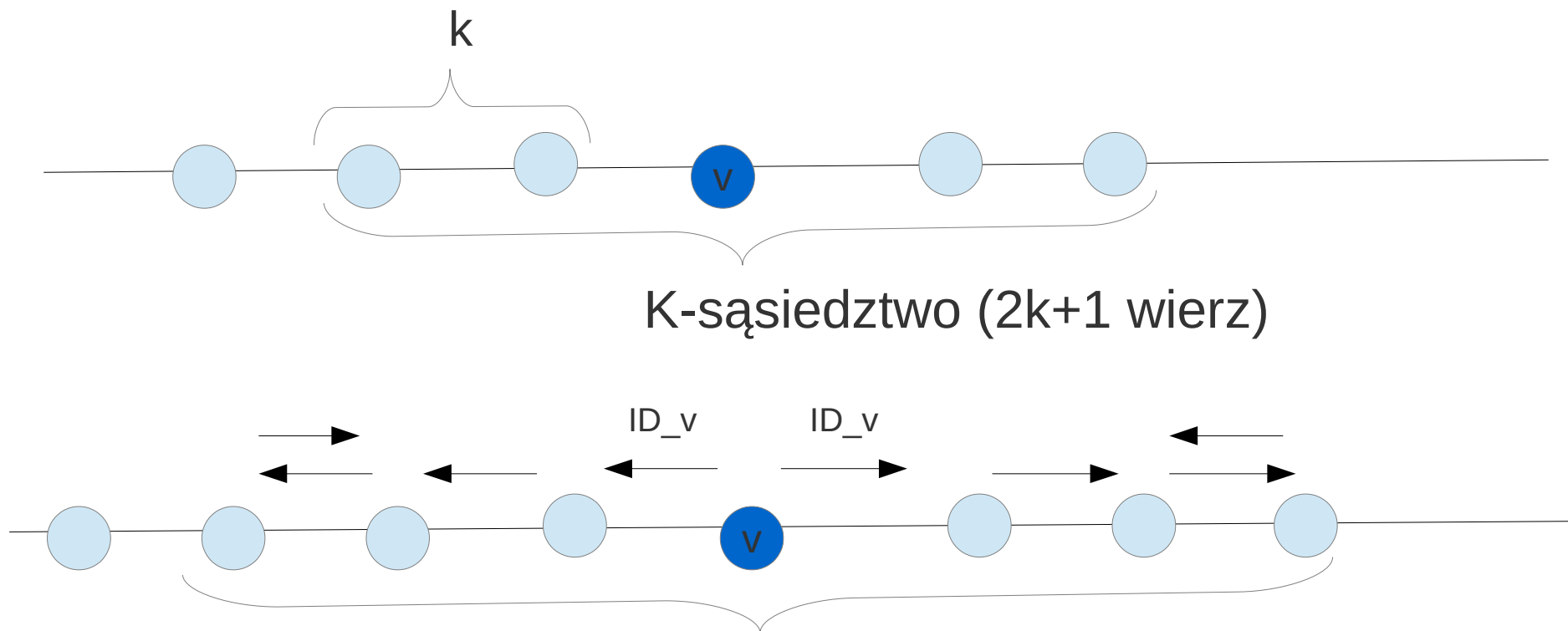
**Pytania:**

1. dlaczego alg wybiera lidera?
2. Ile kom się wysyła?





Problem: leader election, **LE**, ring (zorient), asynch, są ID,  
podamy alg, w którym licz kom =  $O(n \log n)$



Mamy **fazy**: 0, 1, 2, ...

Wierz uczestniczące w  $L$ -tej fazie wykonują powyższe w  $2^L$ - sąsiedztwie  
wierz  $v$  wysyła  $ID_v$  w obie strony + **zasady jak z LE**  $O(n^2)$

Jeśli wierz otrzyma odpowiedź z obu stron to ogłasza się:  
„tymczasowym liderem  $L$ -tej fazy”

W fazie  $L+1$  uczestniczą tymczasowi liderzy  $L$ -tej fazy

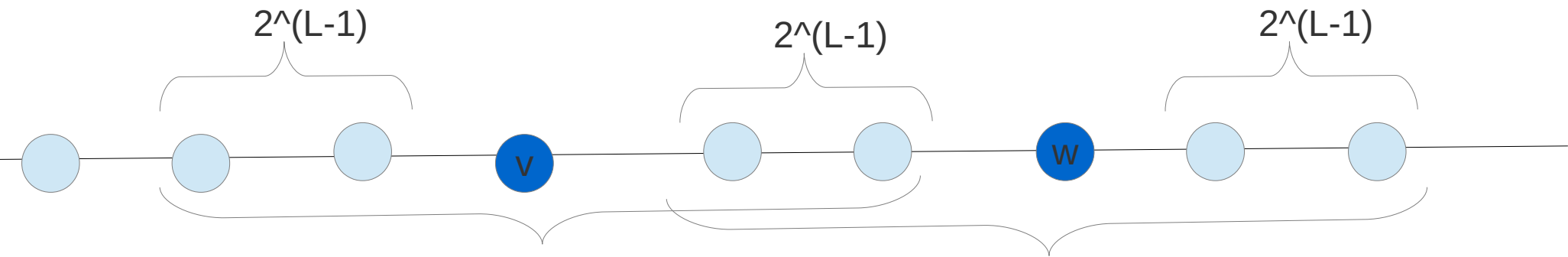
# $LE, O(n \log n)$ kom, c.d.

## Lemat 2.3.1:

Dla  $L \geq 1$ , liczba wierz uczestn w L-tej fazie  $< n/2^{(L-1)}$

## Dowód:

jeśli wierz uczestniczy w L-tej fazie to znaczy że był tymczas liderem (L-1)- fazy,  
tj  $ID_v$  jest max w  $2^{(L-1)}$ -sąsiedztwie „v”



*Liderzy nie mogą być blisko siebie !!!*

Skoro każdy wierz uczestn w fazie L wysyła  $4 \cdot 2^L$  kom,  
to w L-tej fazie łącznie wysyła się  $4 \cdot 2^L \cdot n/2^{(L-1)} = 8n$  kom.  
Liczba faz wynosi  $\log n$  (dlaczego ??)  
stąd złożoność komunikatowa algortymu to  $O(n \log n)$



