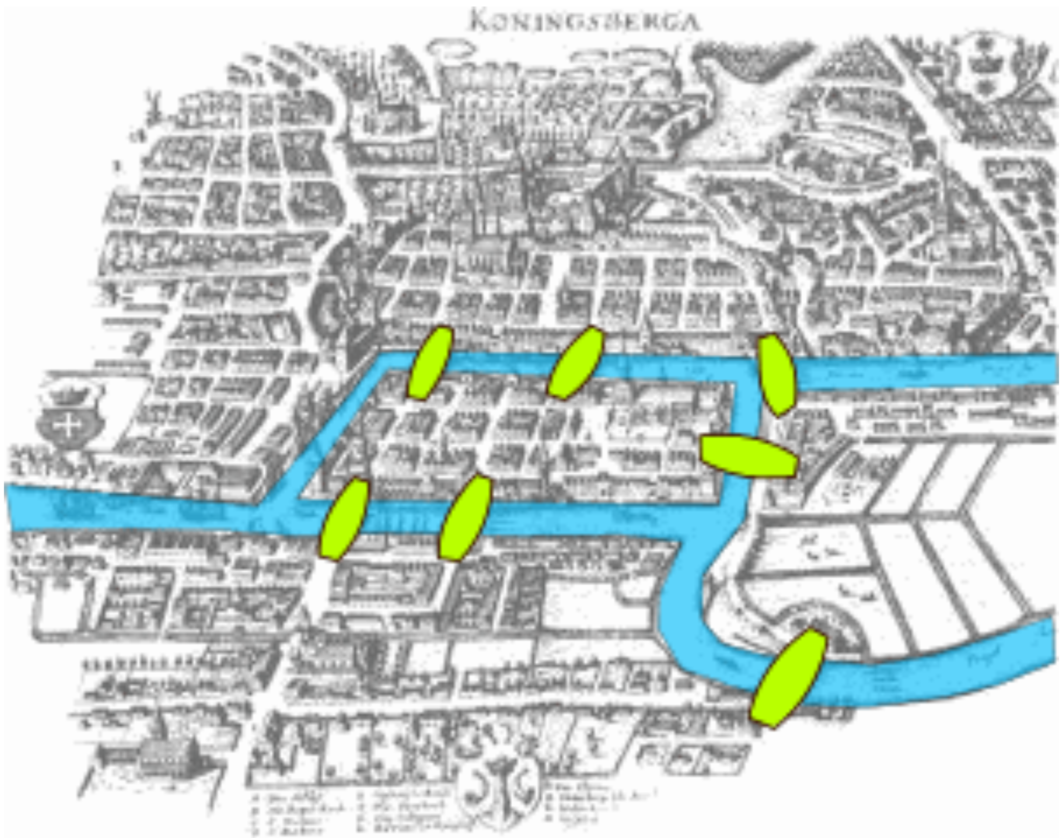
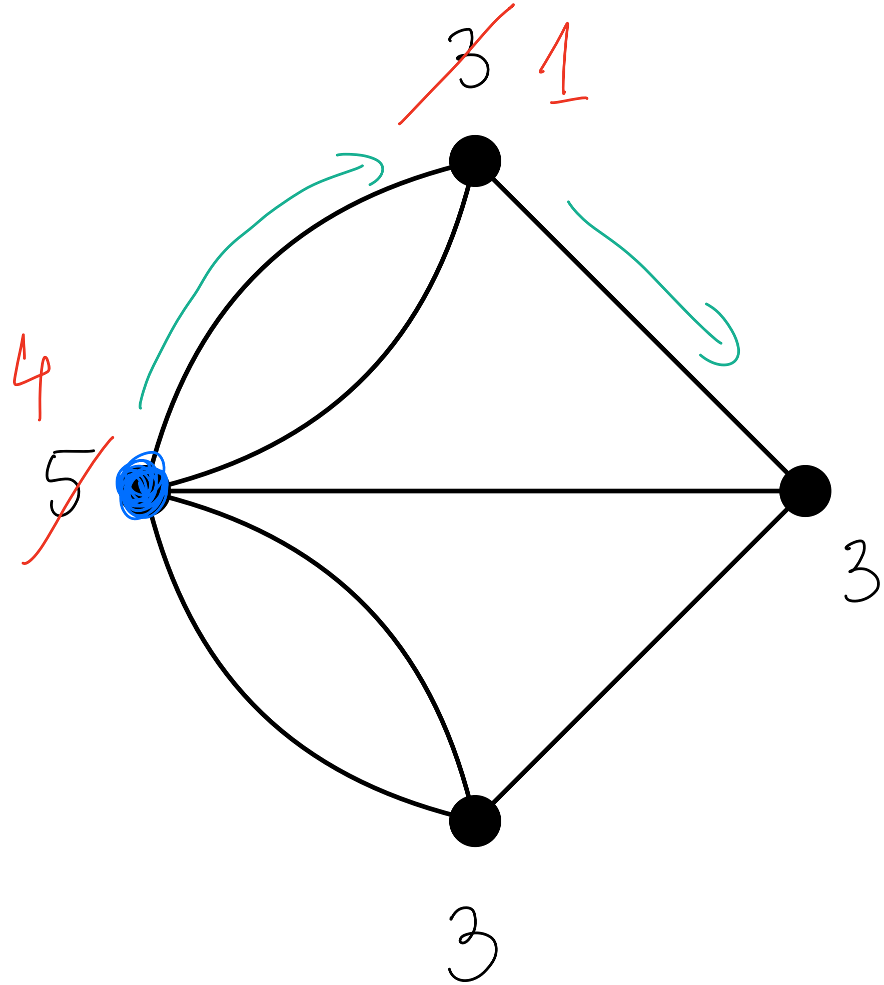
Mosty królewieckie





Grafy eulerowskie

- **Droga Eulera**: droga przechodząca przez wszystkie krawędzie.

  dokładnie raz

- **Obchód Eulera**: zamknięta droga Eulera.
- **Graf eulerowski**: graf posiadający obchód Eulera.

cykl Eulera

Grafy eulerowskie

- **Droga Eulera**: droga przechodząca przez wszystkie krawędzie.
- **Obchód Eulera**: zamknięta droga Eulera.
- **Graf eulerowski**: graf posiadający obchód Eulera.

Twierdzenie

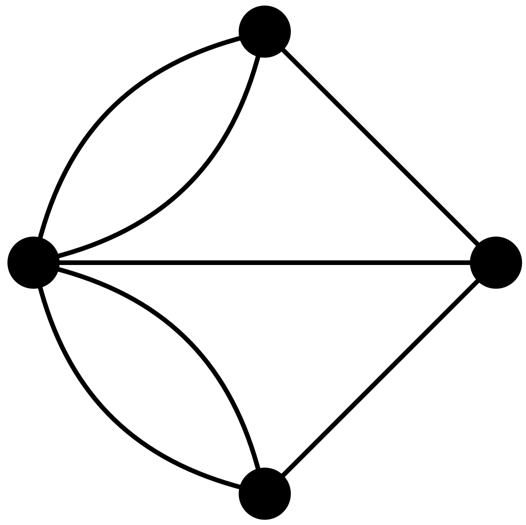
W grafie eulerowskim każdy wierzchołek jest stopnia parzystego.

Grafy eulerowskie

- **Droga Eulera**: droga przechodząca przez wszystkie krawędzie.
- **Obchód Eulera**: zamknięta droga Eulera.
- **Graf eulerowski**: graf posiadający obchód Eulera.

Twierdzenie

W grafie eulerowskim każdy wierzchołek jest stopnia parzystego.

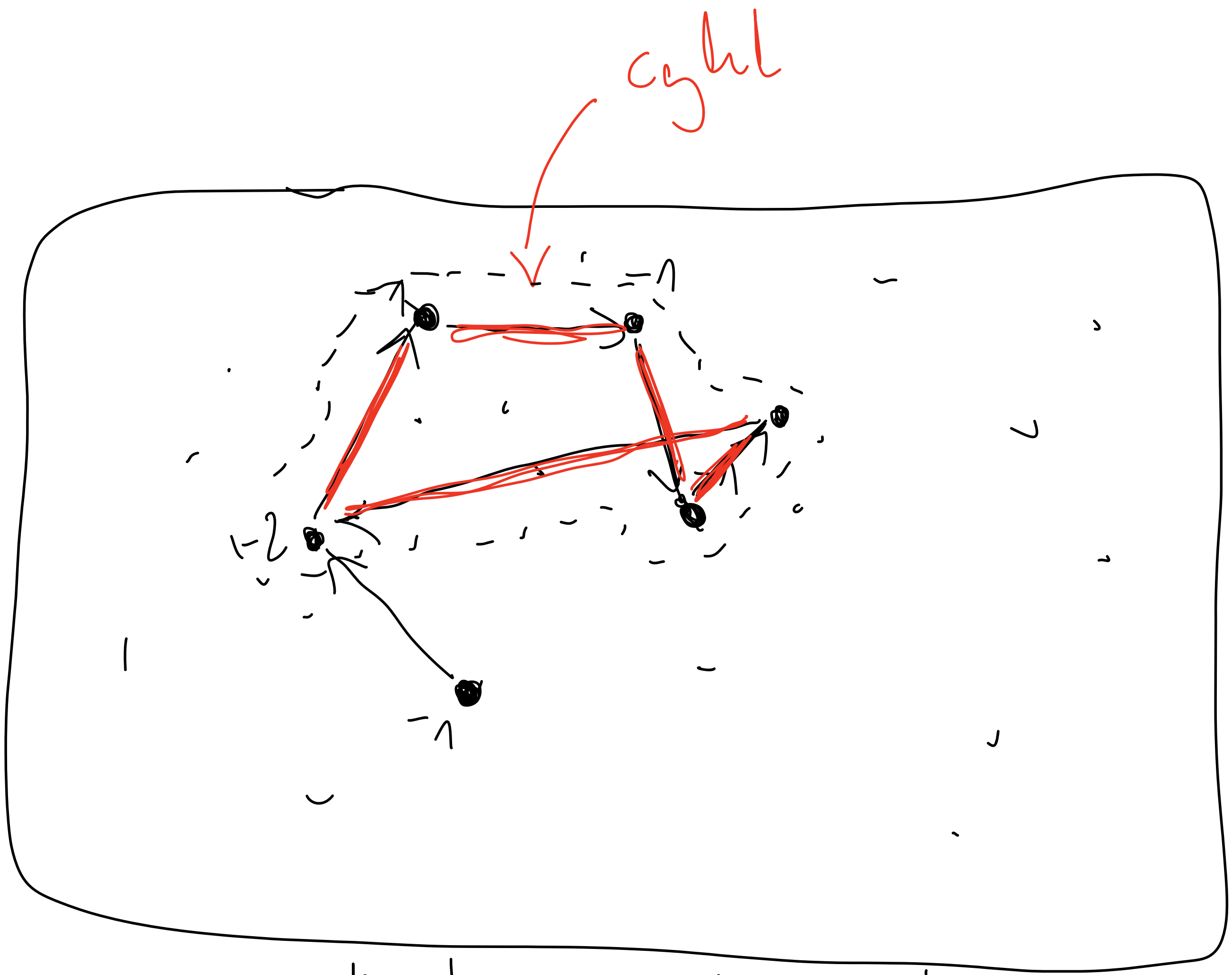


Twierdzenie Eulera

Graf jest eulerowski wtedy i tylko wtedy, gdy jest spójny i każdy jego wierzchołek ma stopień parzysty.

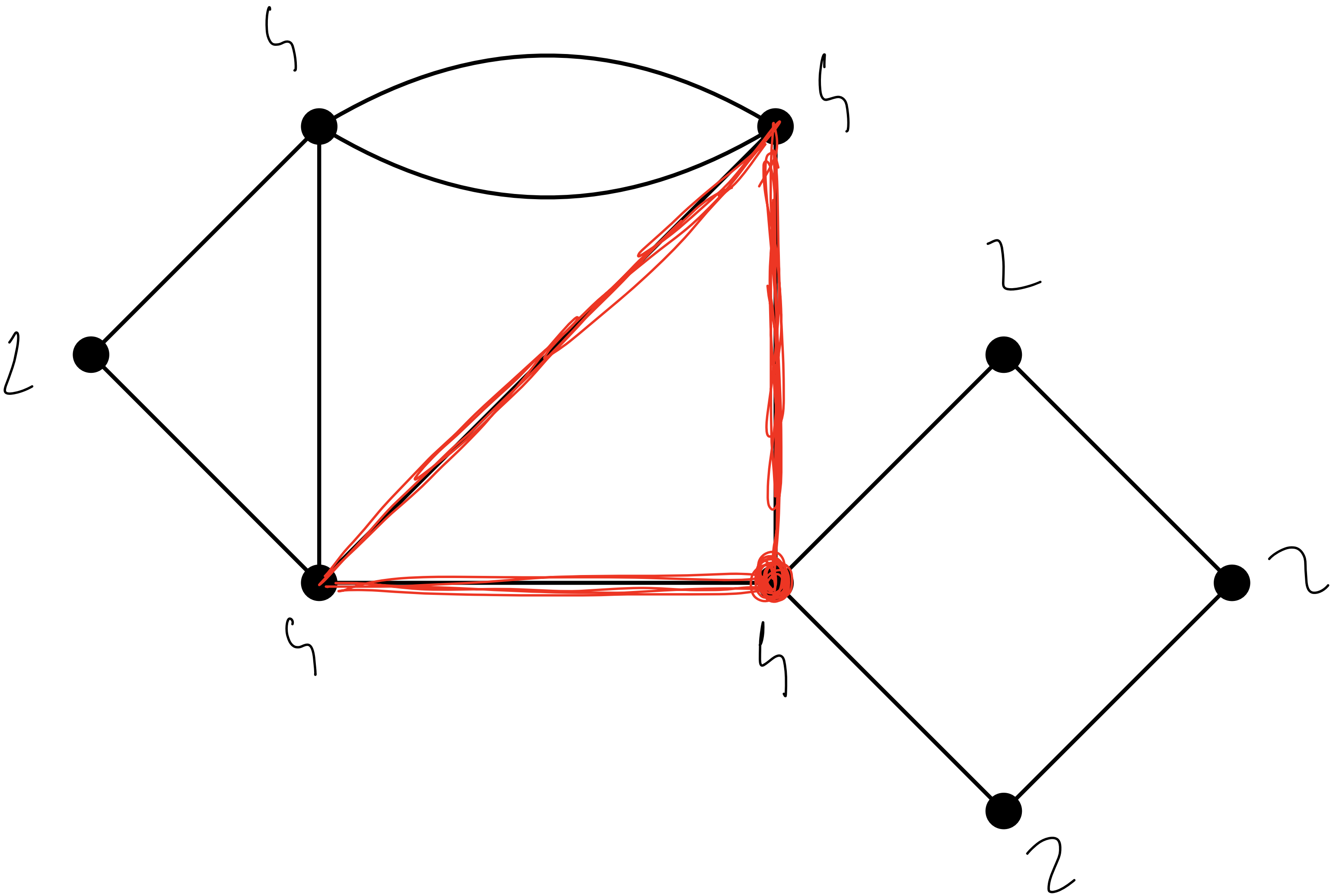
($\Rightarrow$) ✓

($\Leftarrow$) ?

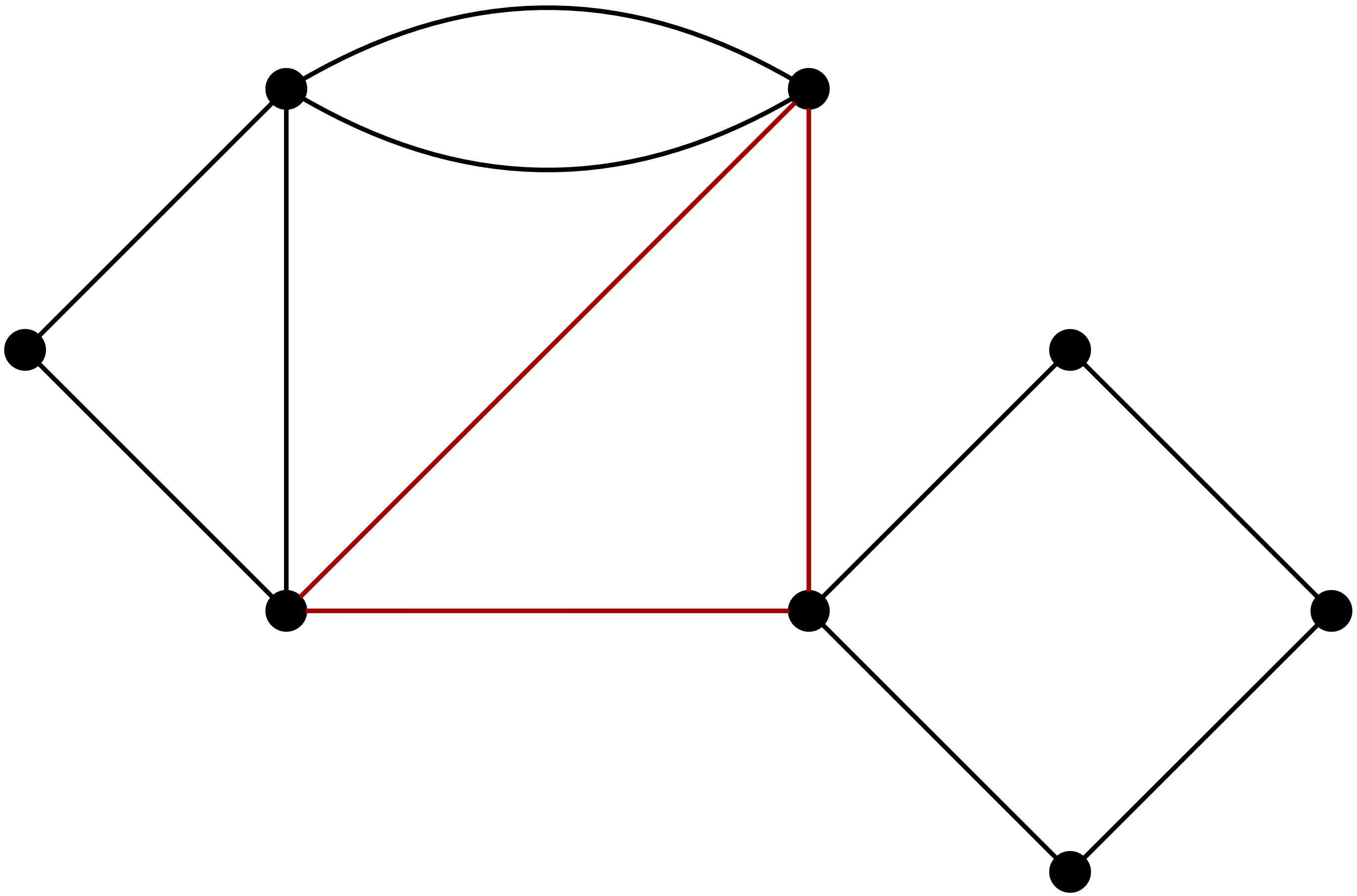


Indukcja po liczbie krawędzi.

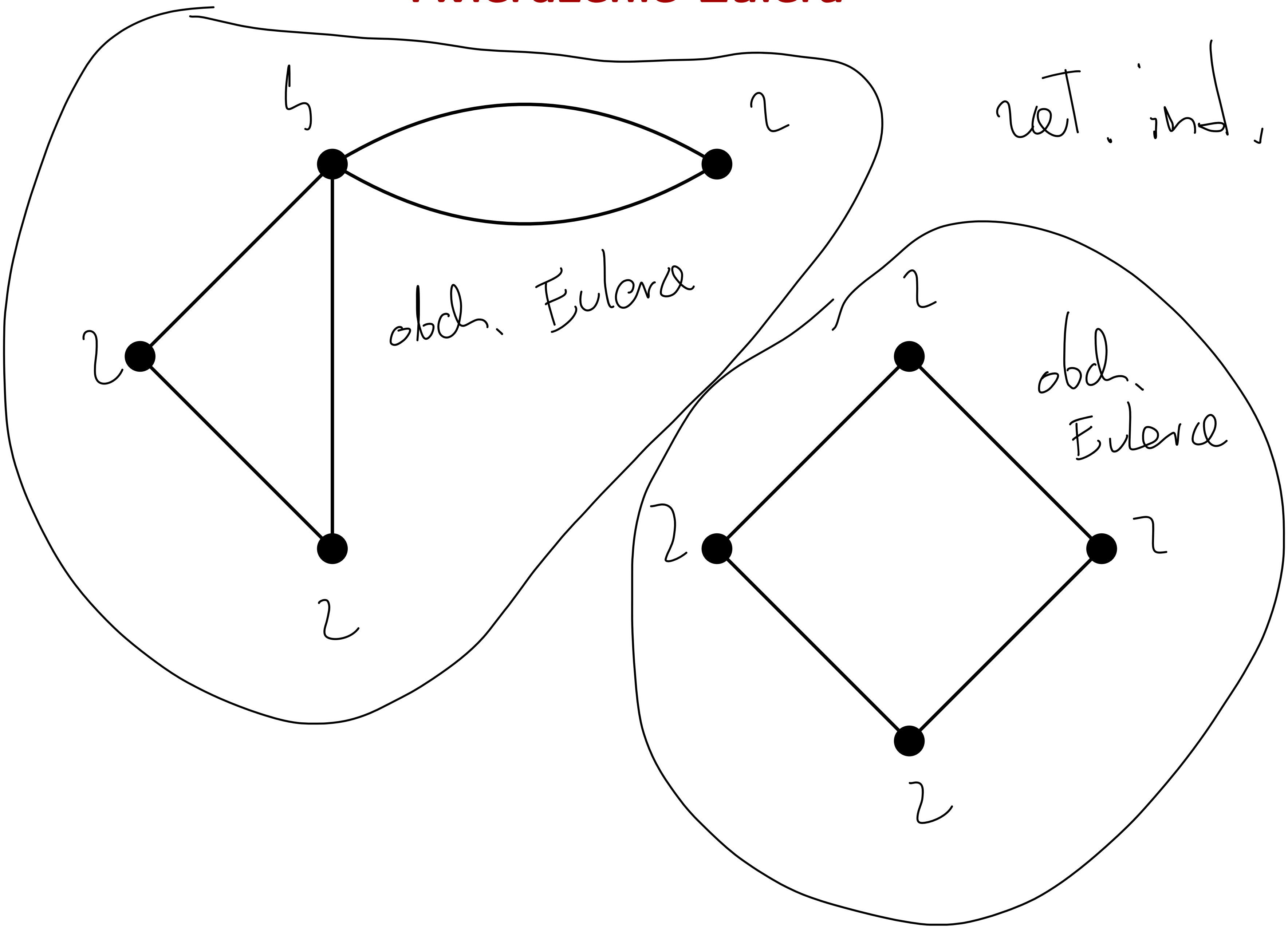
Twierdzenie Eulera



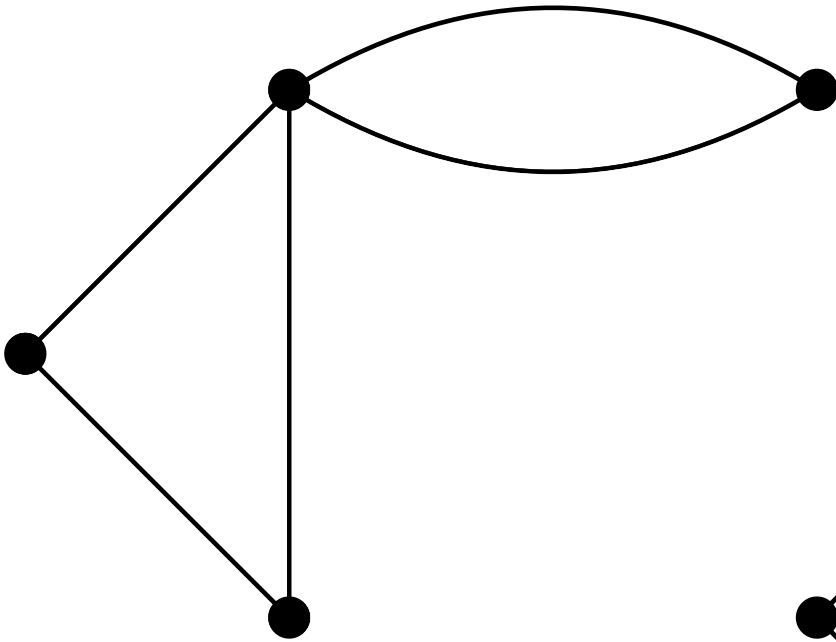
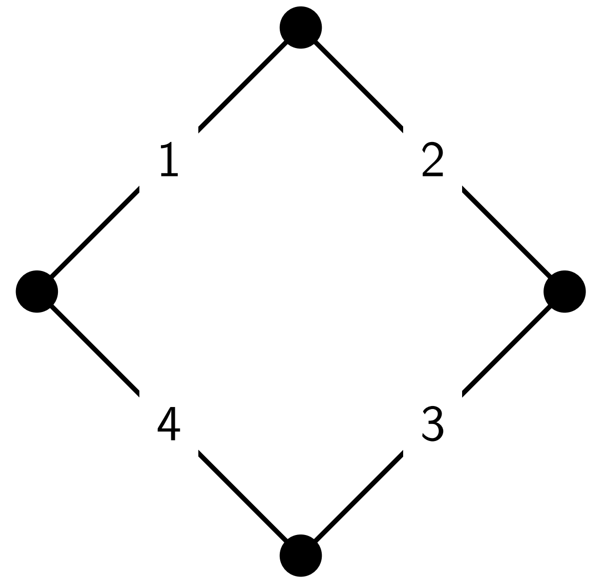
Twierdzenie Eulera



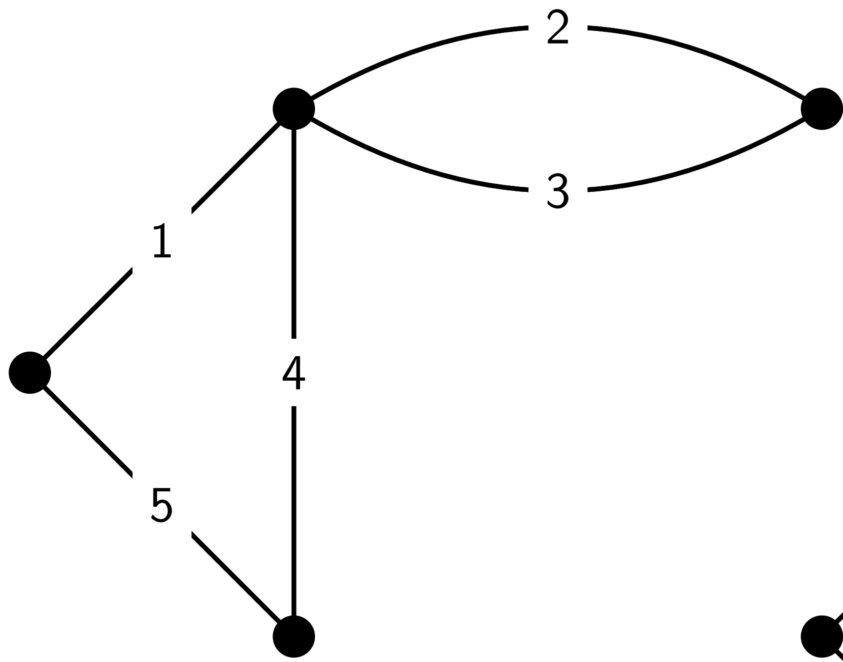
Twierdzenie Eulera

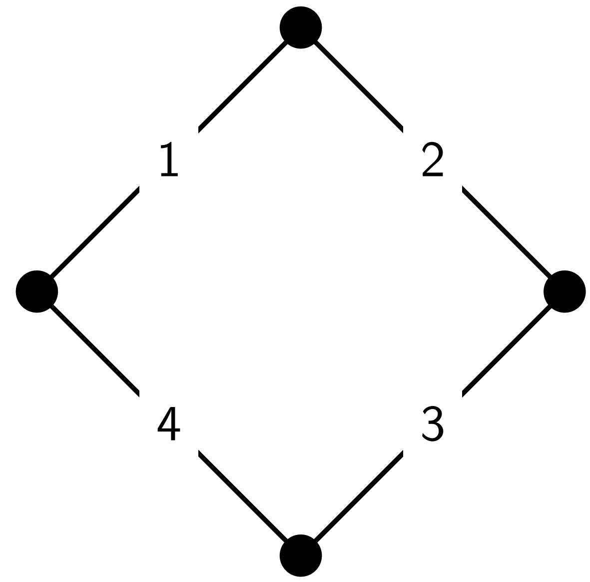


Twierdzenie Eulera

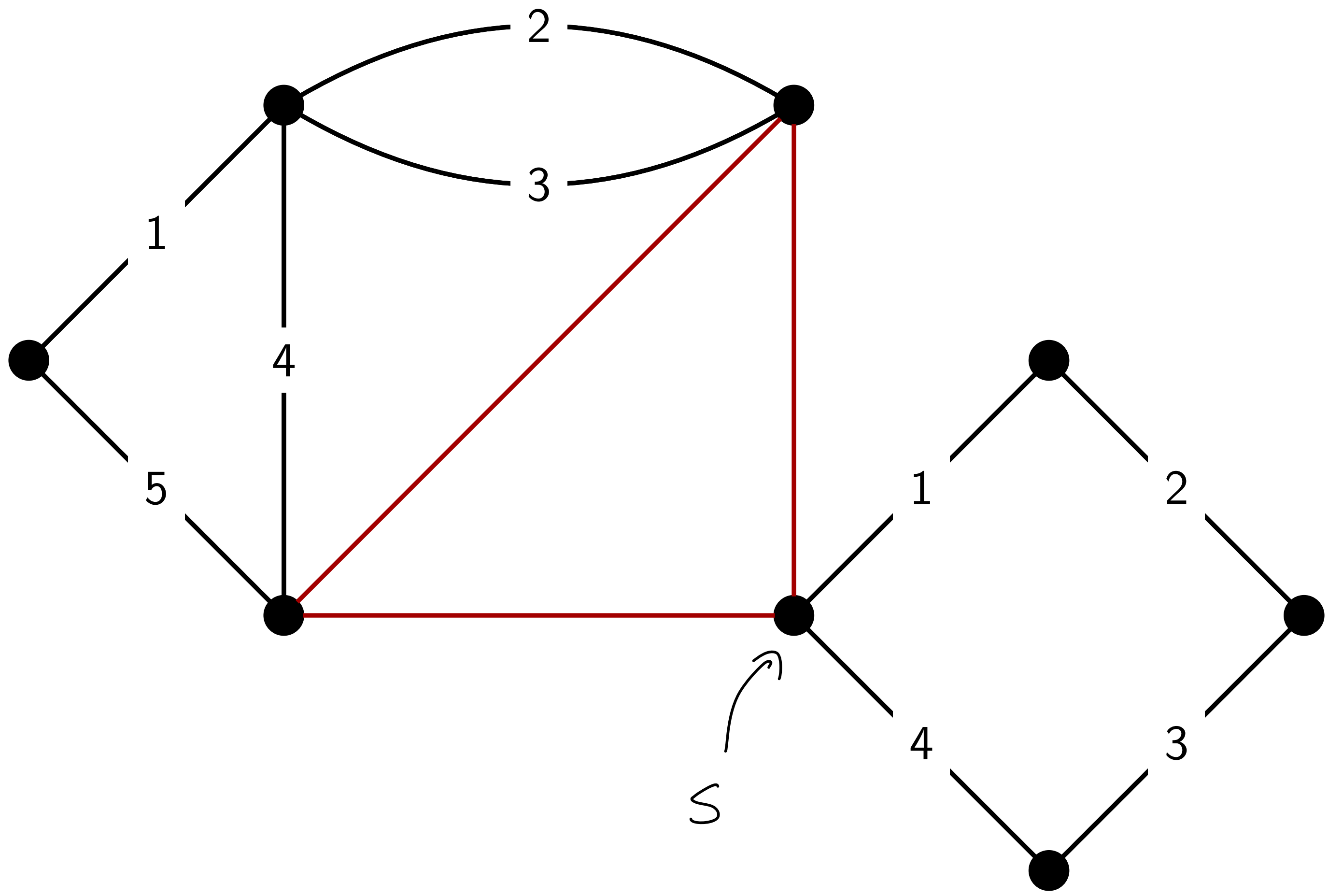



Twierdzenie Eulera

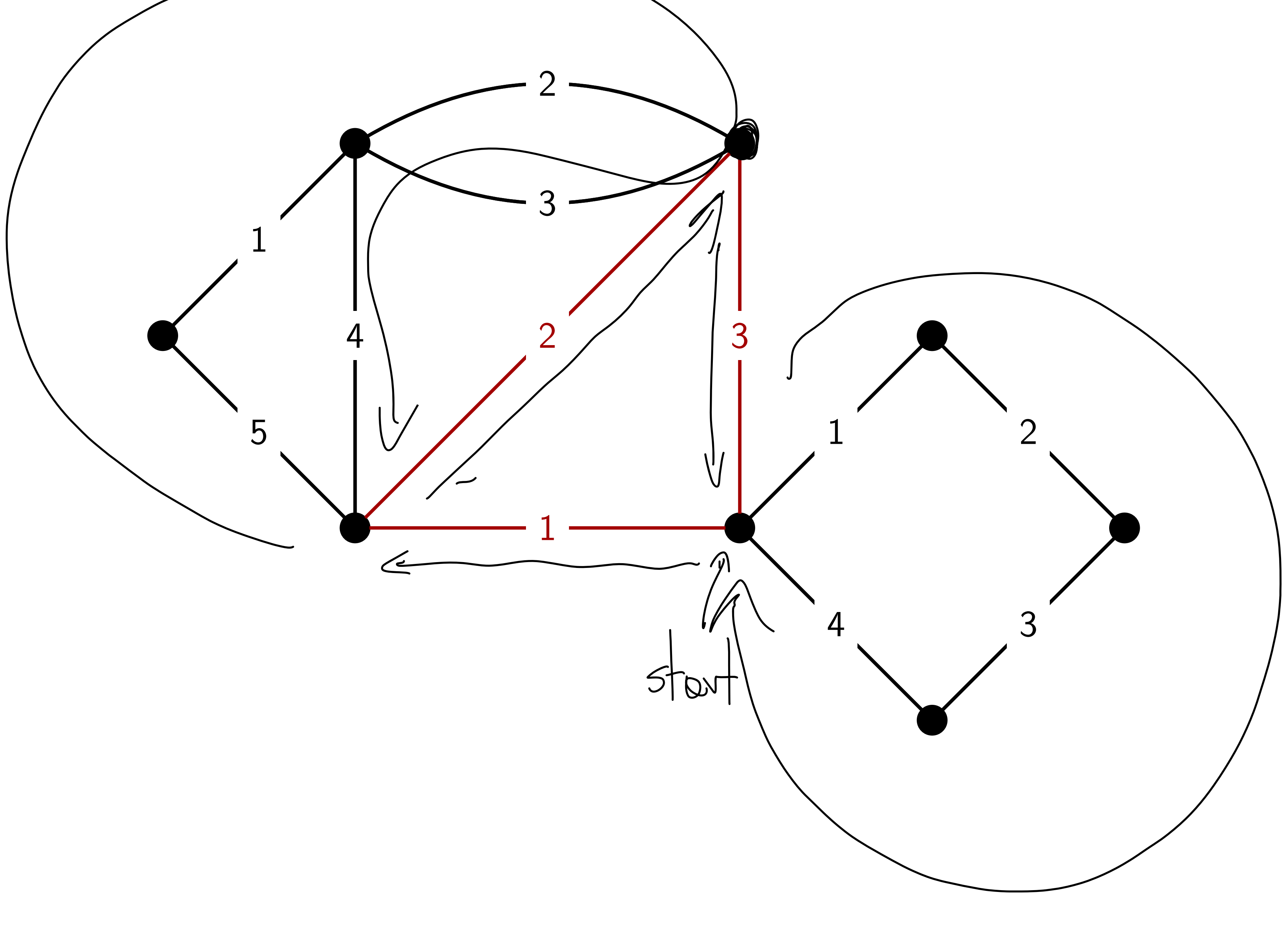




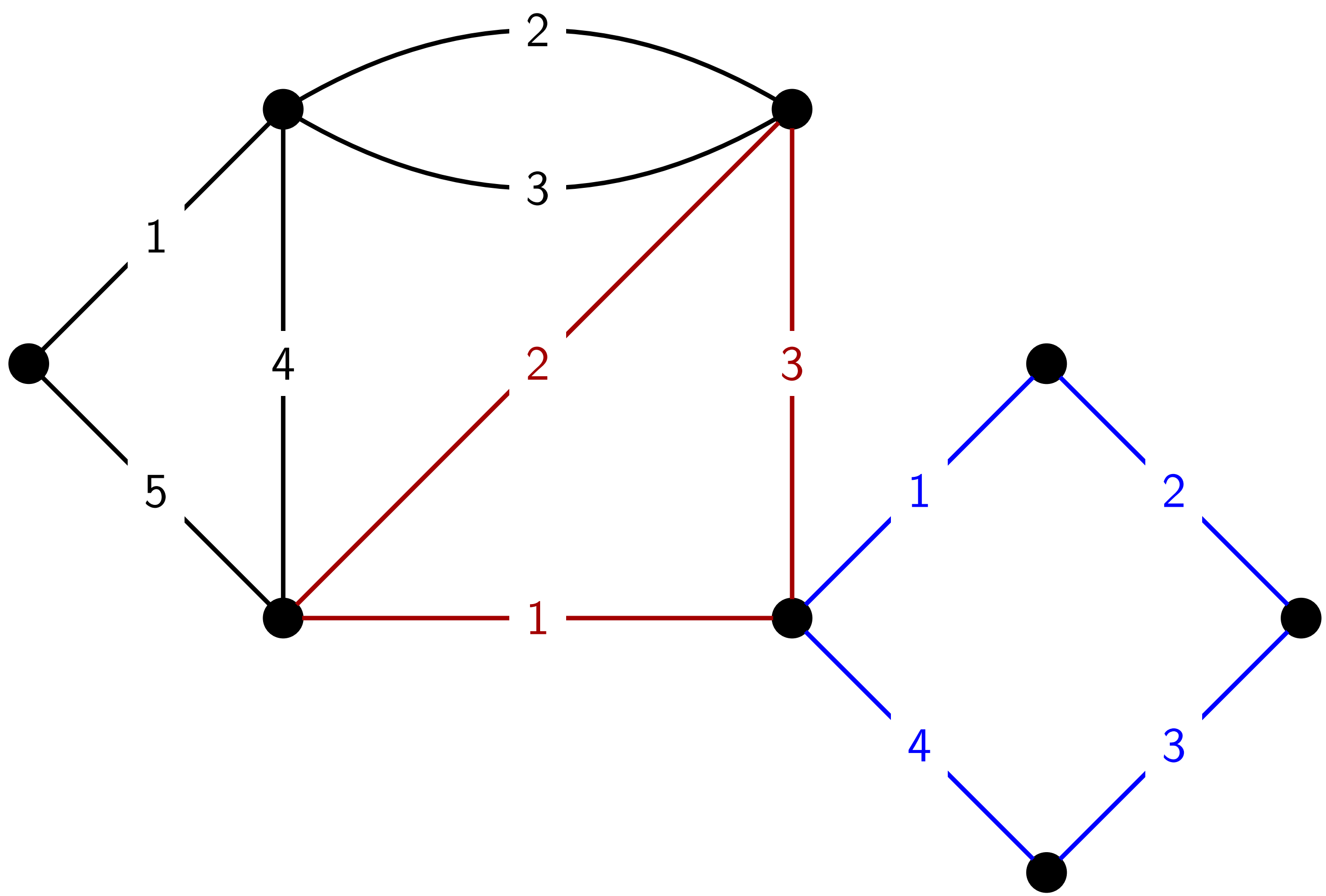
Twierdzenie Eulera



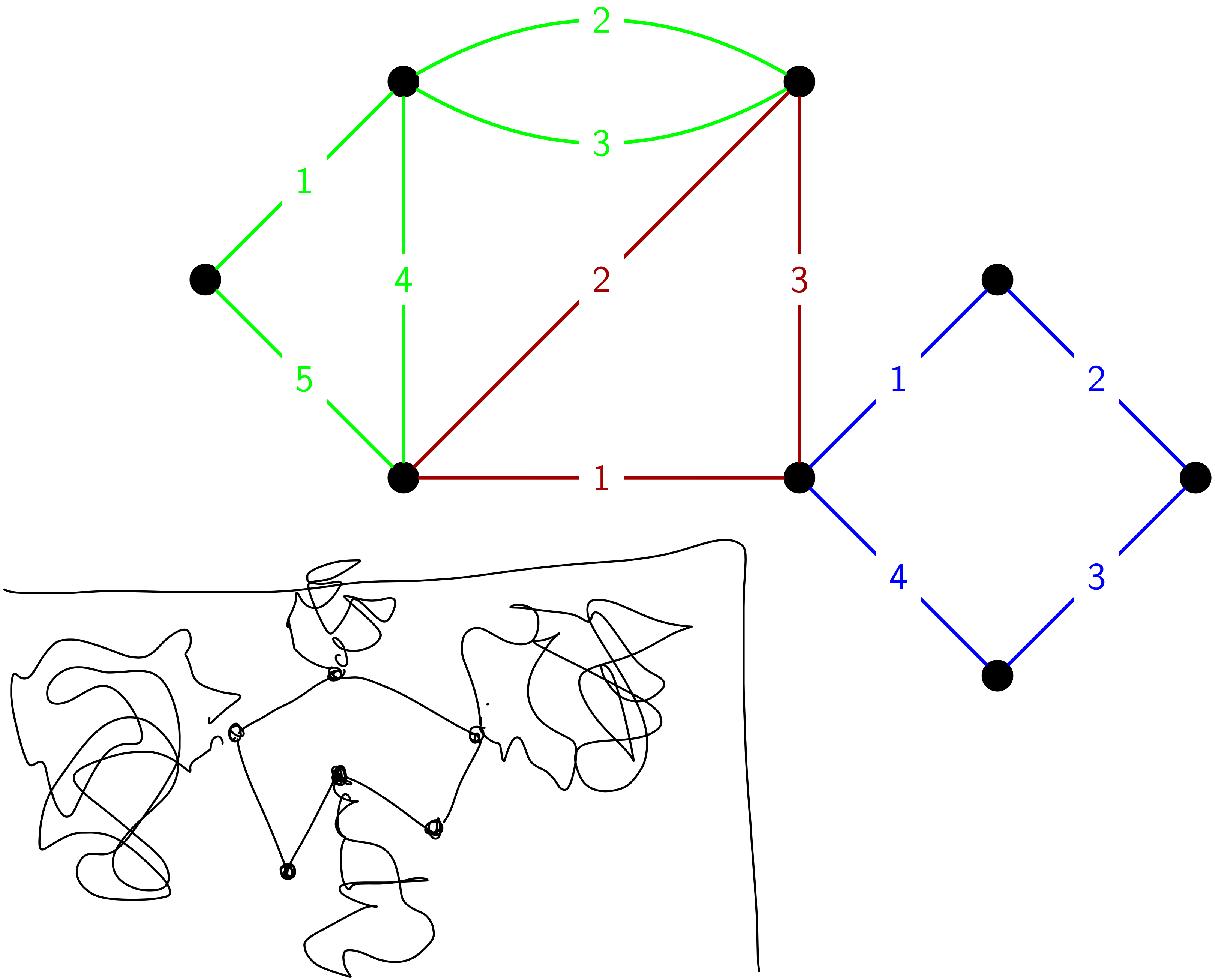
Twierdzenie Eulera



Twierdzenie Eulera



Twierdzenie Eulera

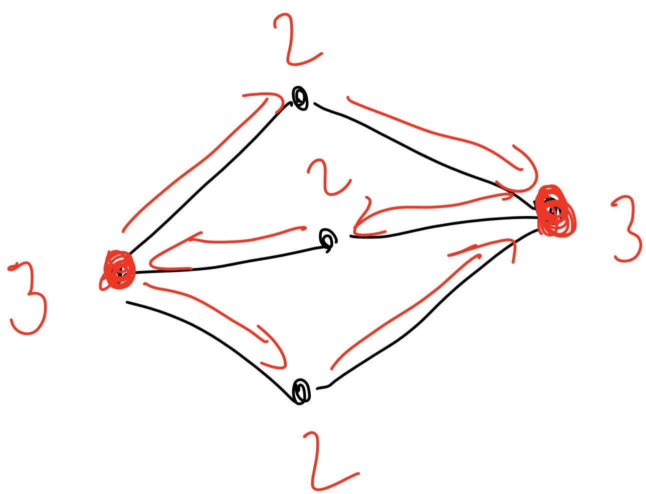


Algorytm Fleury'ego

1. E_E: poszukiwany ciąg krawędzi.

Algorytm Fleury'ego

1. E_E: poszukiwany ciąg krawędzi.
2. Jeżeli w grafie istnieje jakiś wierzchołek v stopnia nieparzystego, to go wybierz. Jeżeli taki wierzchołek nie istnieje, to wybierz dowolny wierzchołek v.



Algorytm Fleury'ego

1. E_E: poszukiwany ciąg krawędzi.
2. Jeżeli w grafie istnieje jakiś wierzchołek v stopnia nieparzystego, to go wybierz. Jeżeli taki wierzchołek nie istnieje, to wybierz dowolny wierzchołek v.
   - $\rightsquigarrow$ Jeżeli z wierzchołka v nie wychodzi żadna krawędź, to przerwij.

Algorytm Fleury'ego

1. E_E: poszukiwany ciąg krawędzi.
2. Jeżeli w grafie istnieje jakiś wierzchołek v stopnia nieparzystego, to go wybierz. Jeżeli taki wierzchołek nie istnieje, to wybierz dowolny wierzchołek v.
   - Jeżeli z wierzchołka v nie wychodzi żadna krawędź, to przerwij.
   - Jeżeli pozostała dokładnie jedna krawędź $e = vw$ wychodząca z wierzchołka v do w, to usuń ten wierzchołek i tę krawędź.

Algorytm Fleury'ego

1. E_E: poszukiwany ciąg krawędzi.
2. Jeżeli w grafie istnieje jakiś wierzchołek v stopnia nieparzystego, to go wybierz. Jeżeli taki wierzchołek nie istnieje, to wybierz dowolny wierzchołek v.
   - ⇝ Jeżeli z wierzchołka v nie wychodzi żadna krawędź, to przerwij.
   - ⇝ Jeżeli pozostała dokładnie jedna krawędź $e = vw$ wychodząca z wierzchołka v do w, to usuń ten wierzchołek i tę krawędź.
   - ⇝ Jeżeli pozostała więcej niż jedna krawędź wychodząca z v, to wybierz taką krawędź $e = vw$, po usunięciu której graf pozostanie spójny, a następnie usuń tę krawędź.

Algorytm Fleury'ego

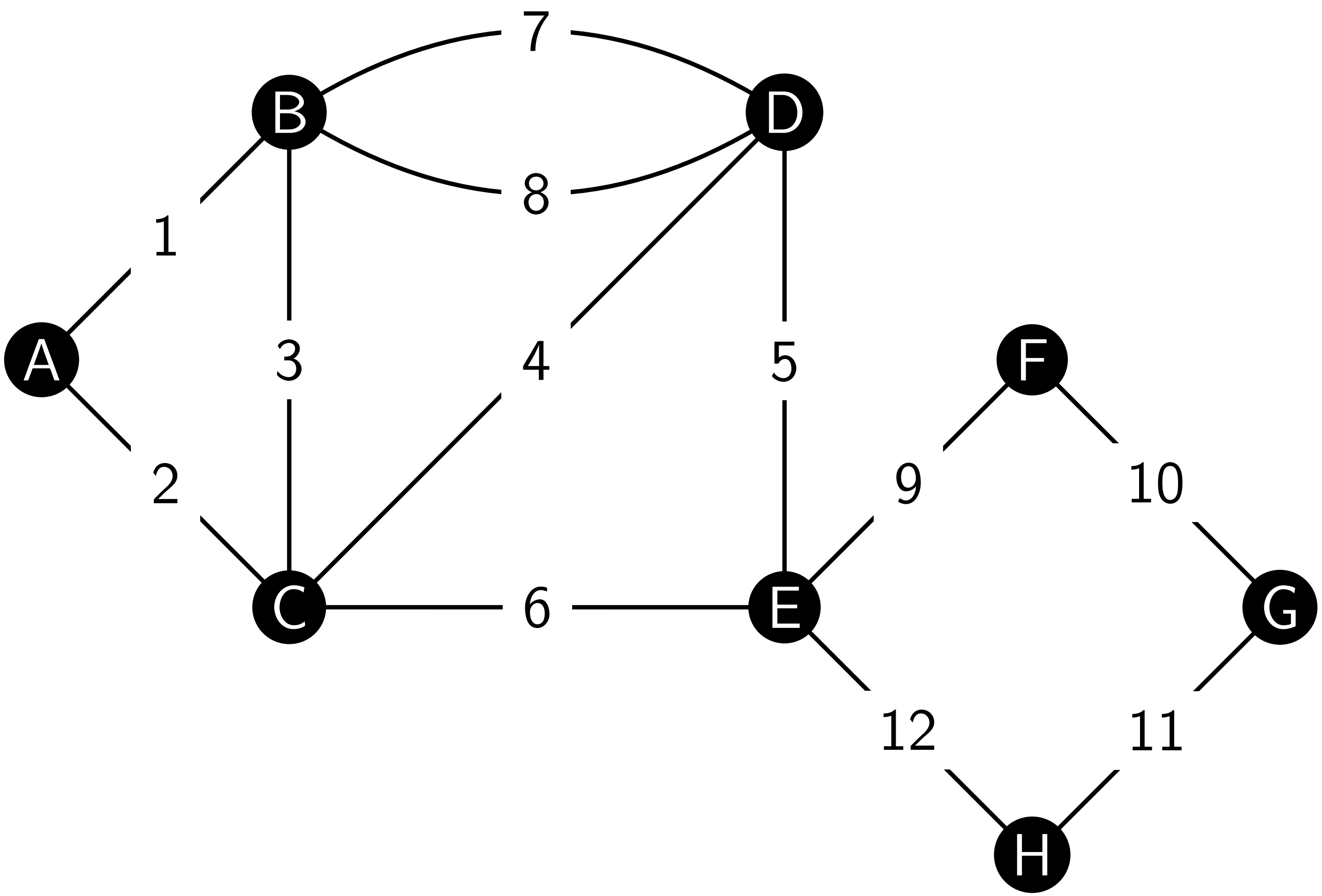
1. E_E: poszukiwany ciąg krawędzi.
2. Jeżeli w grafie istnieje jakiś wierzchołek v stopnia nieparzystego, to go wybierz. Jeżeli taki wierzchołek nie istnieje, to wybierz dowolny wierzchołek v.
   - ⇝ Jeżeli z wierzchołka v nie wychodzi żadna krawędź, to przerwij.
   - ⇝ Jeżeli pozostała dokładnie jedna krawędź $e = vw$ wychodząca z wierzchołka v do w, to usuń ten wierzchołek i tę krawędź.
   - ⇝ Jeżeli pozostała więcej niż jedna krawędź wychodząca z v, to wybierz taką krawędź $e = vw$, po usunięciu której graf pozostanie spójny, a następnie usuń tę krawędź.
3. Dodaj e do E_E.

Algorytm Fleury'ego

1. E_E: poszukiwany ciąg krawędzi.
2. Jeżeli w grafie istnieje jakiś wierzchołek v stopnia nieparzystego, to go wybierz. Jeżeli taki wierzchołek nie istnieje, to wybierz dowolny wierzchołek v.
   - ⇝ Jeżeli z wierzchołka v nie wychodzi żadna krawędź, to przerwij.
   - ⇝ Jeżeli pozostała dokładnie jedna krawędź $e = vw$ wychodząca z wierzchołka v do w, to usuń ten wierzchołek i tę krawędź.
   - ⇝ Jeżeli pozostała więcej niż jedna krawędź wychodząca z v, to wybierz taką krawędź $e = vw$, po usunięciu której graf pozostanie spójny, a następnie usuń tę krawędź.
3. Dodaj e do E_E.
4. Zastąp v przez w i wróć do kroku 2.

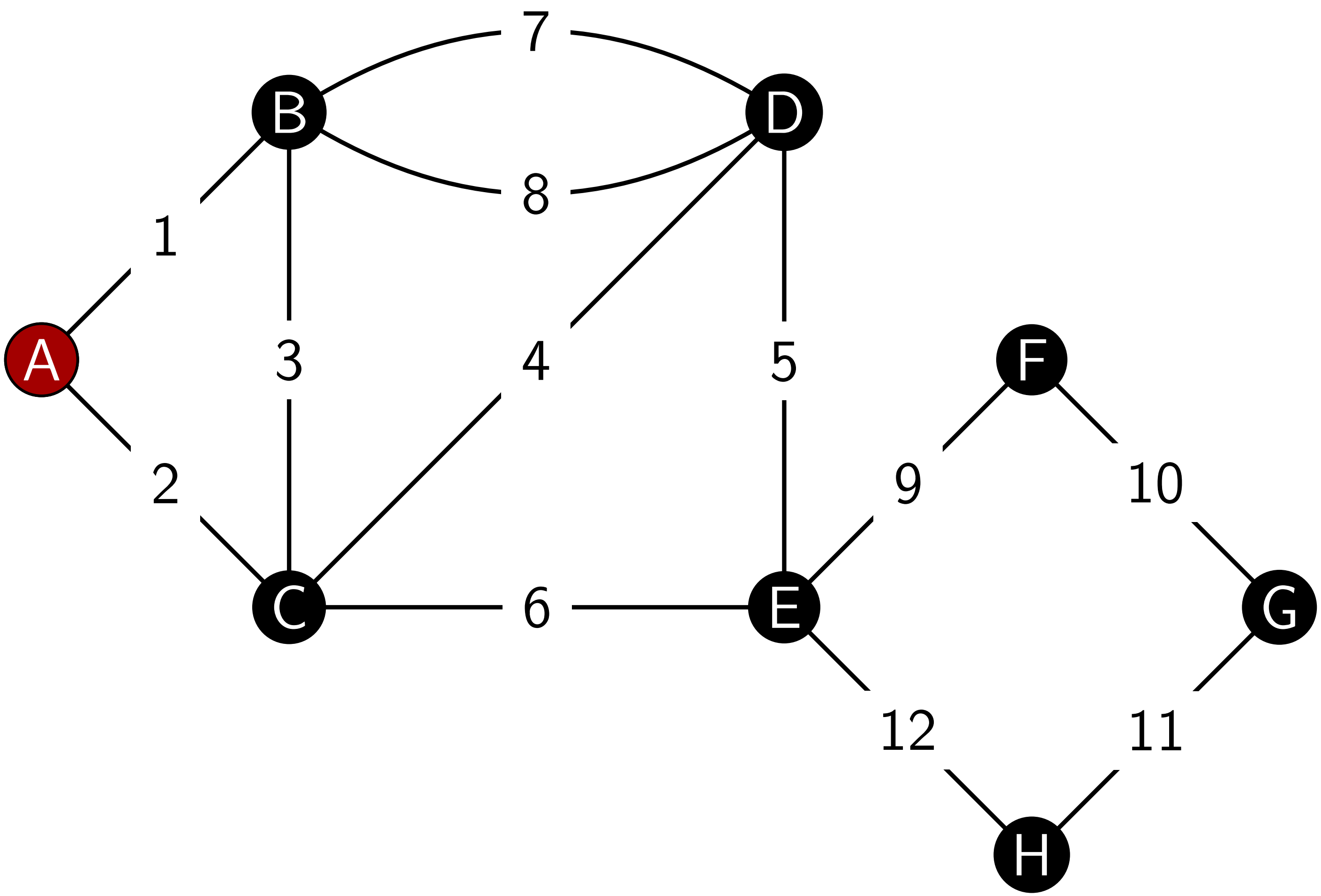
Algorytm Fleury'ego

start



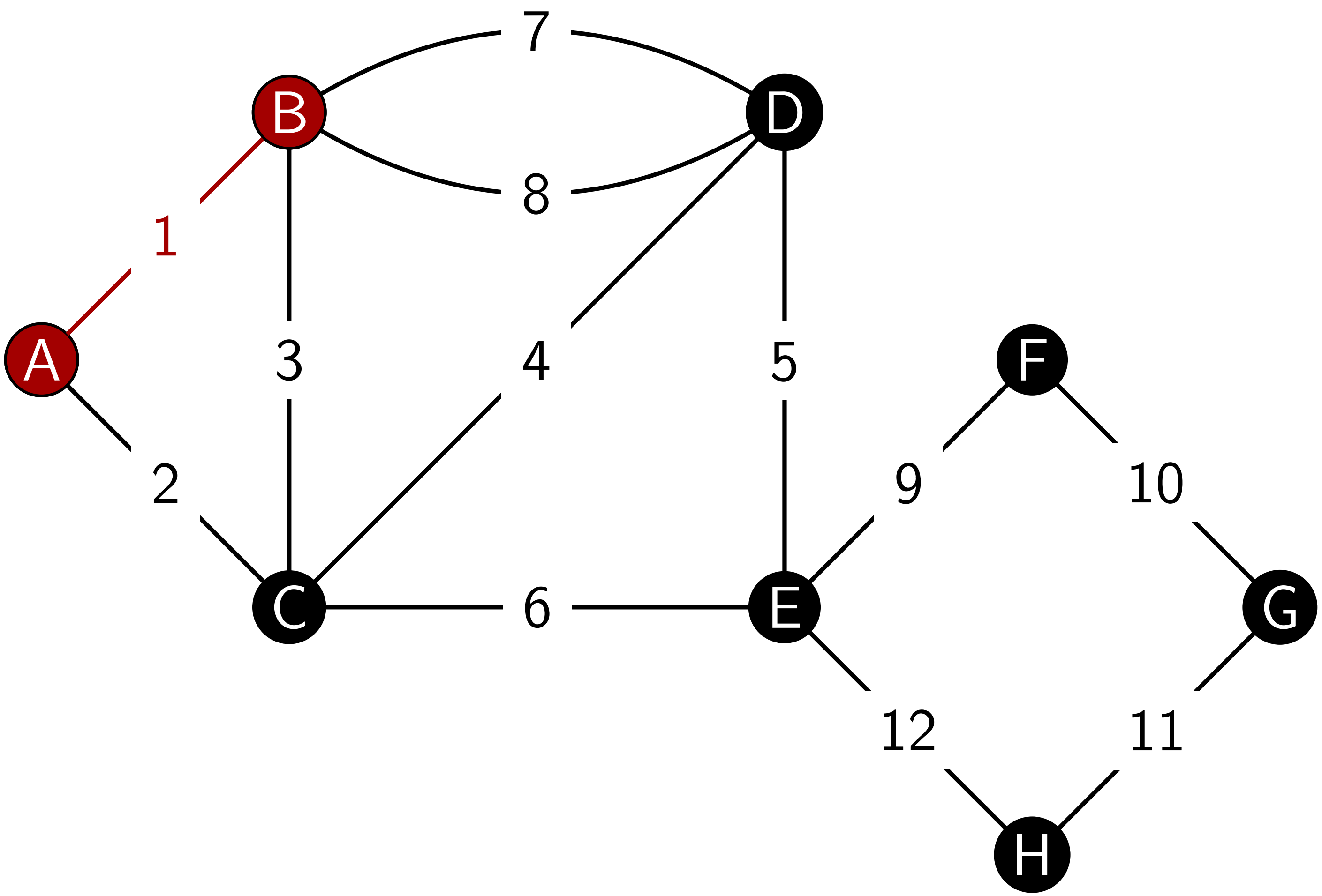
$E_E =$

Algorytm Fleury'ego



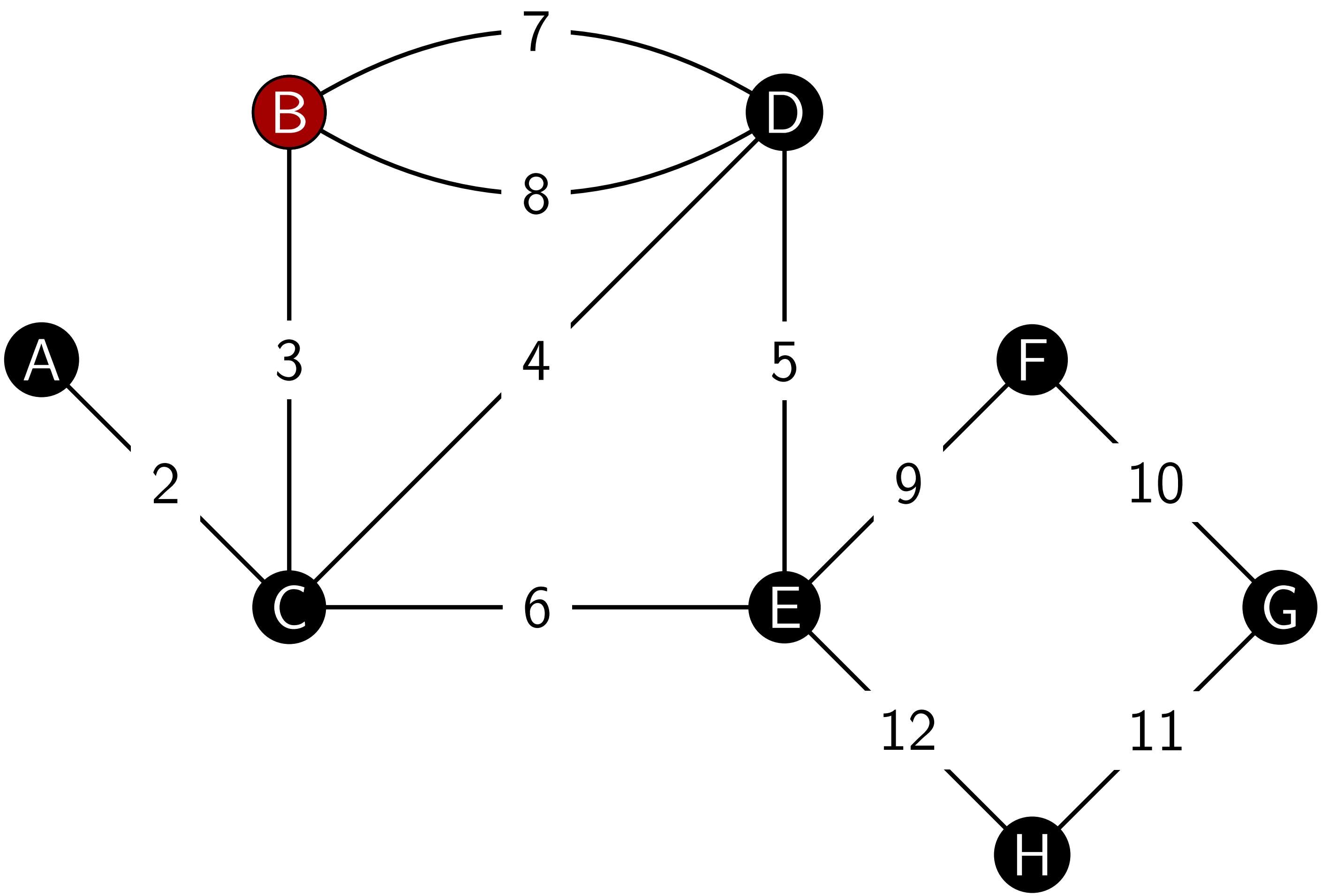
$$E_E =$$

Algorytm Fleury'ego



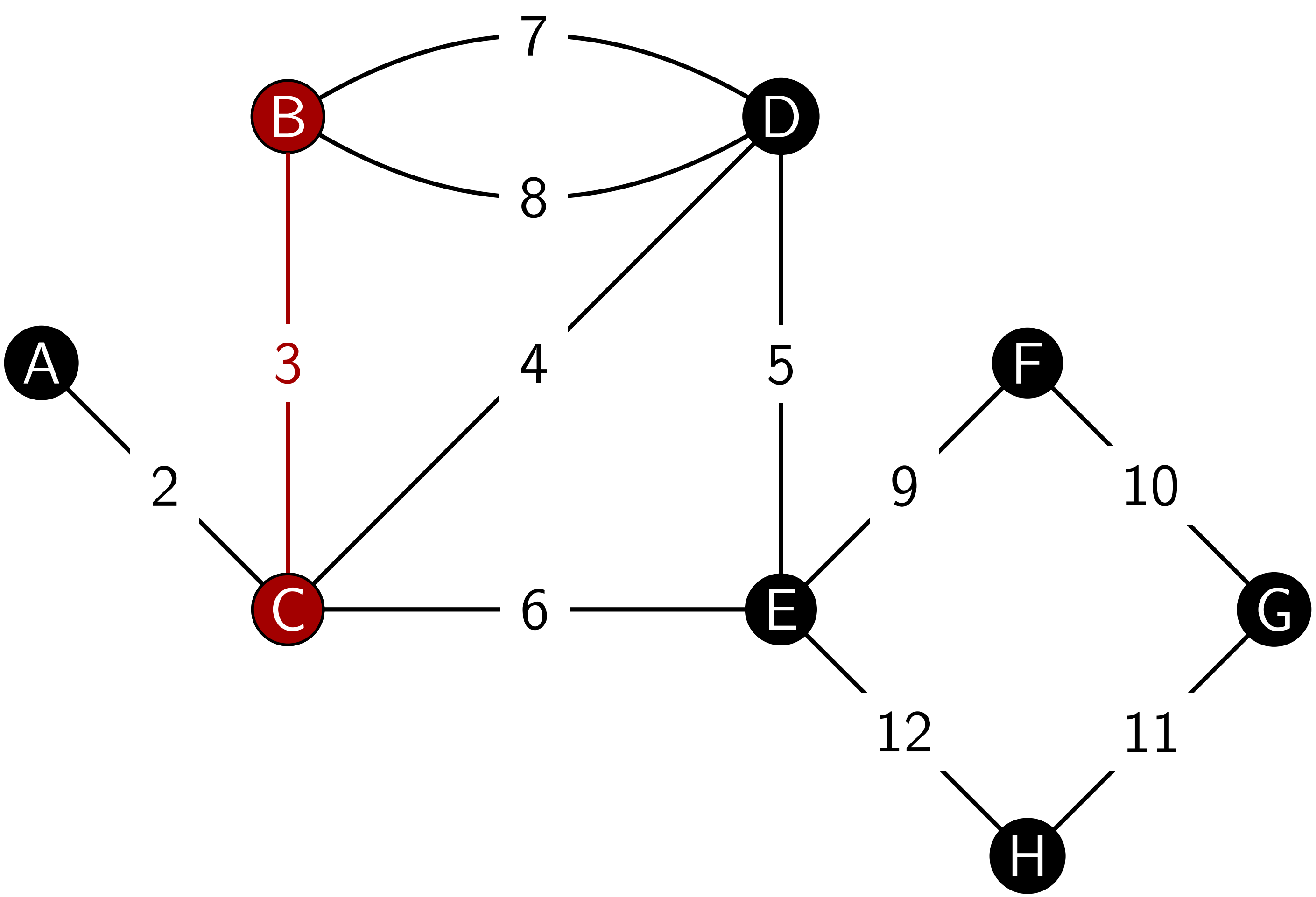
$E_E =$

Algorytm Fleury'ego



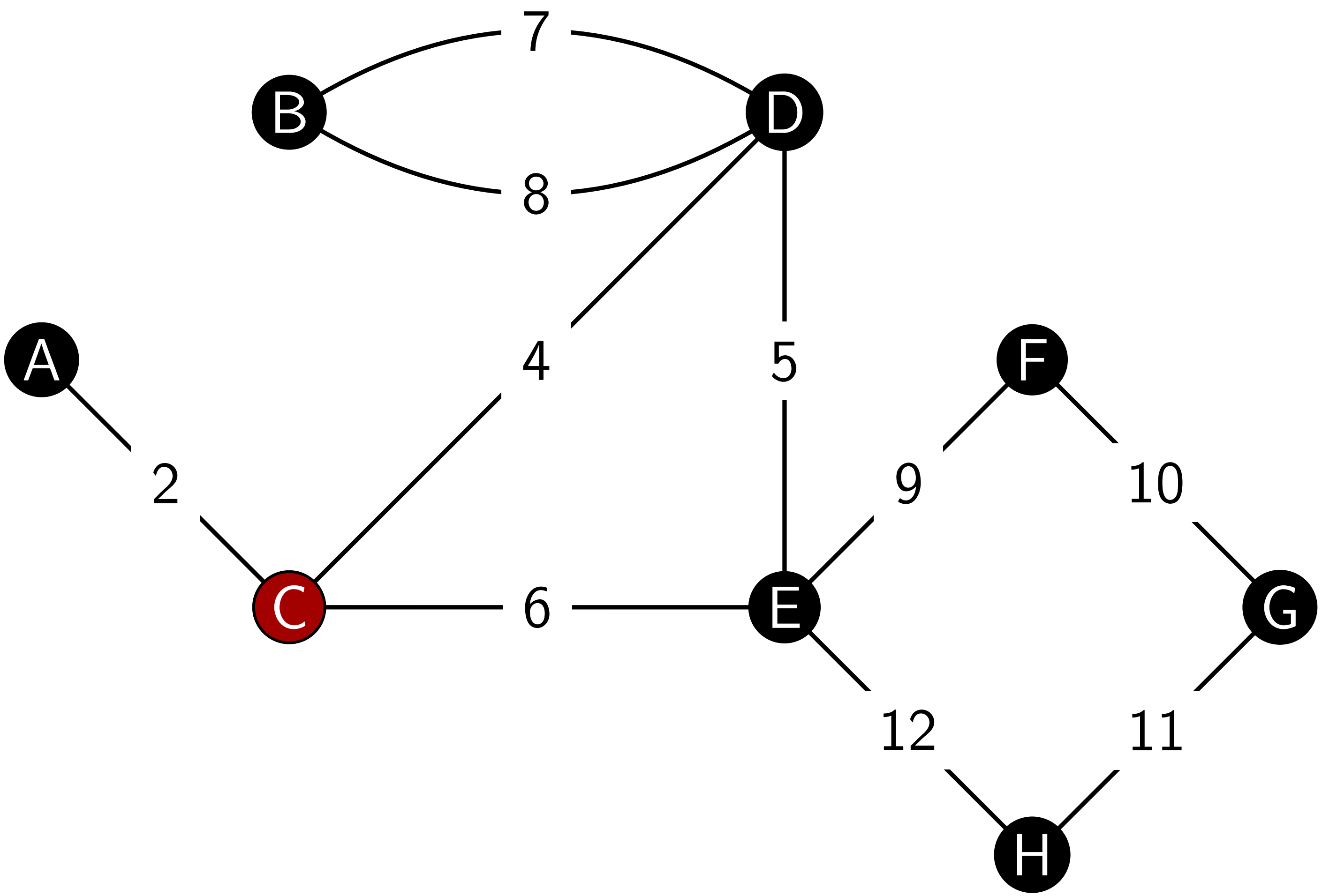
$$E_E = 1$$

Algorytm Fleury'ego



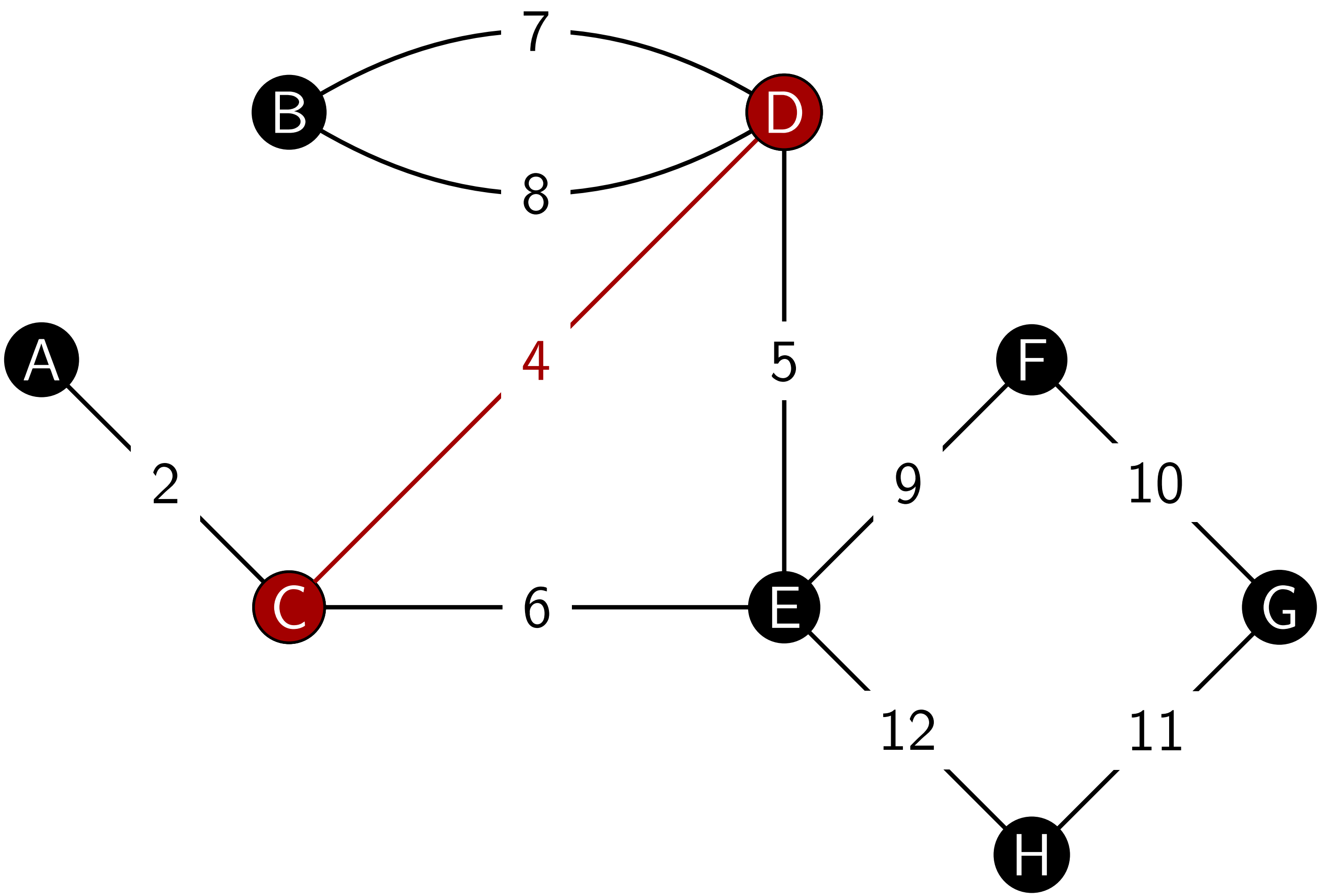
$E_E = 1$

Algorytm Fleury'ego



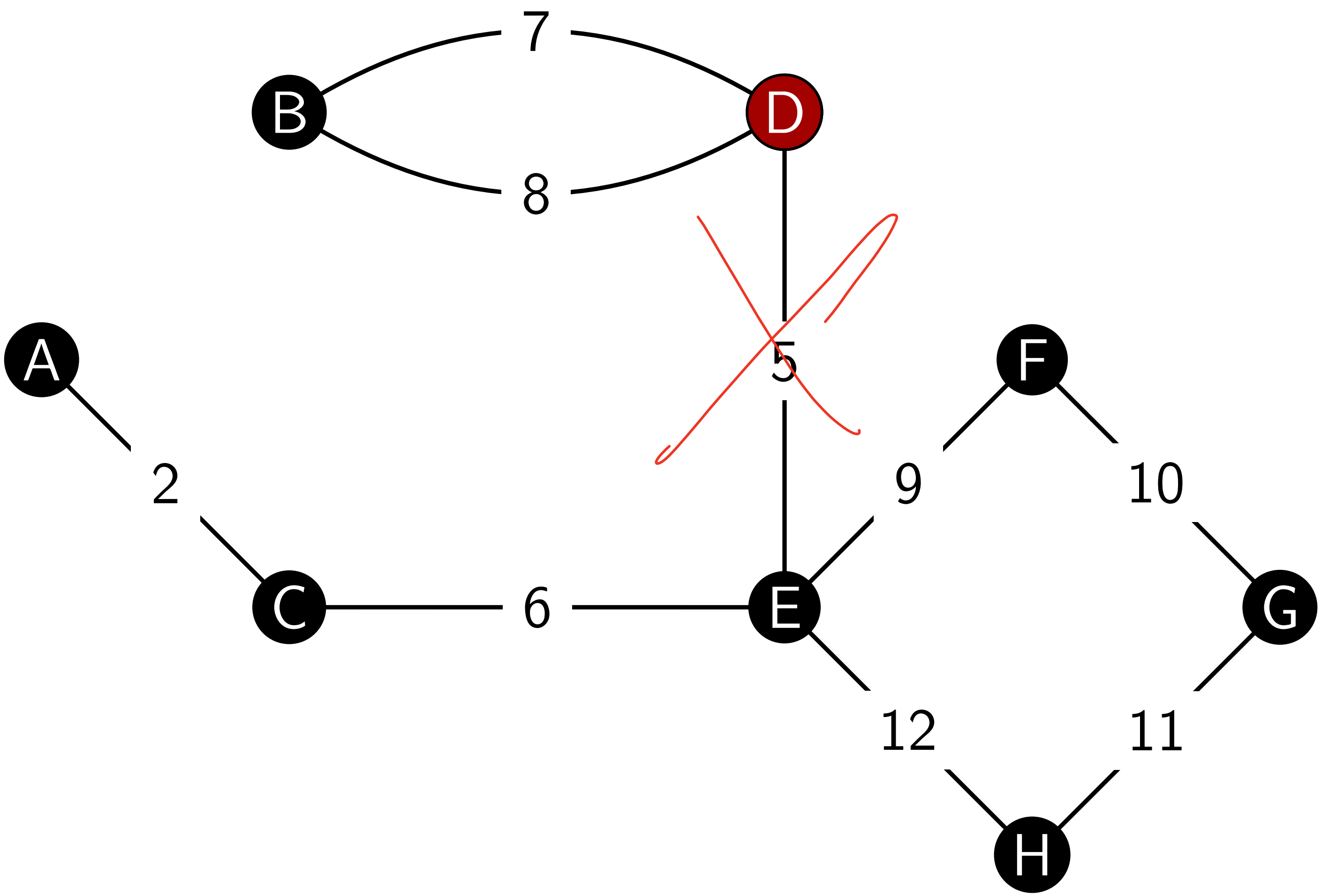
$E_E = 1, 3$

Algorytm Fleury'ego



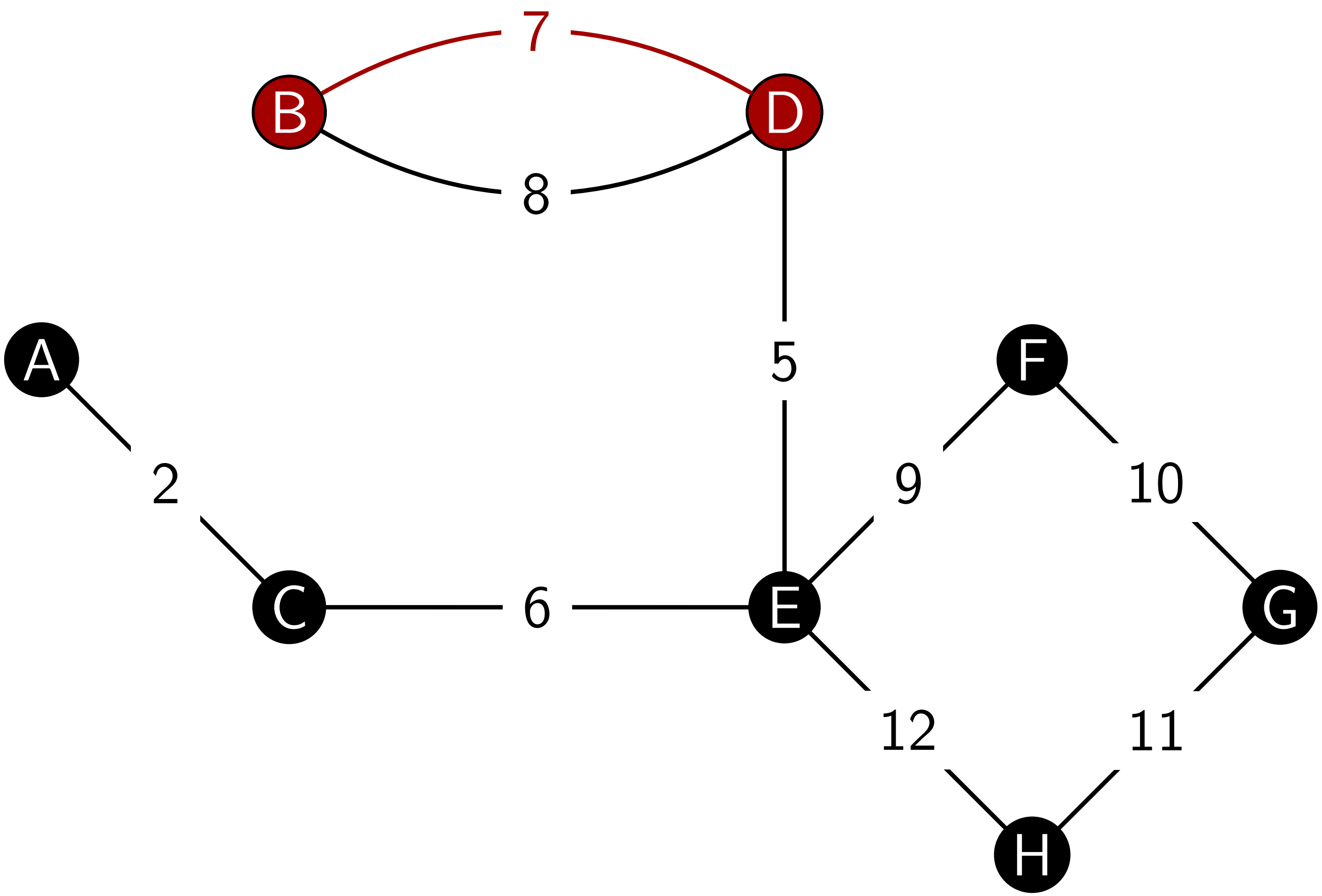
$E_E = 1, 3$

Algorytm Fleury'ego



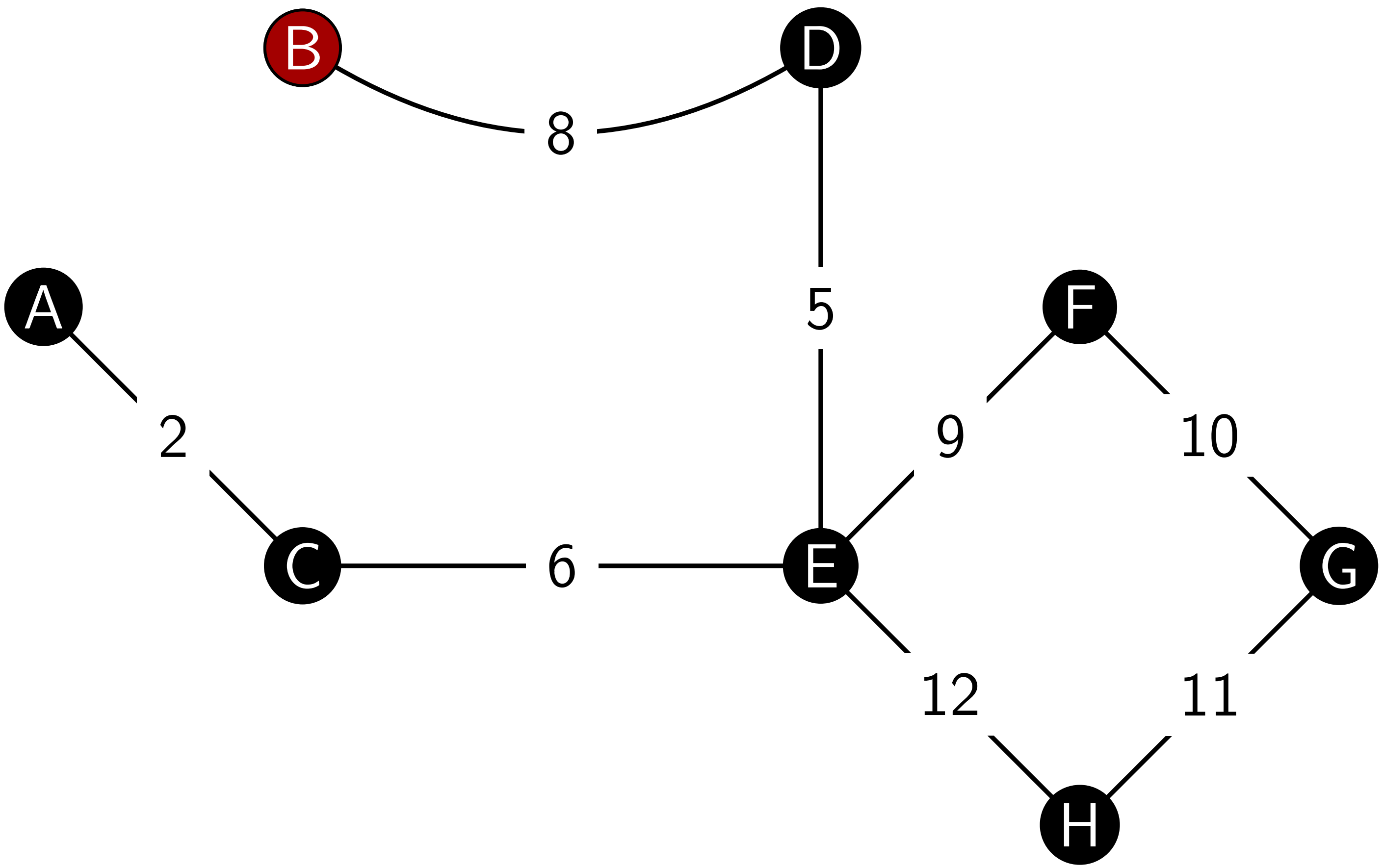
$E_E = 1, 3, 4$

Algorytm Fleury'ego



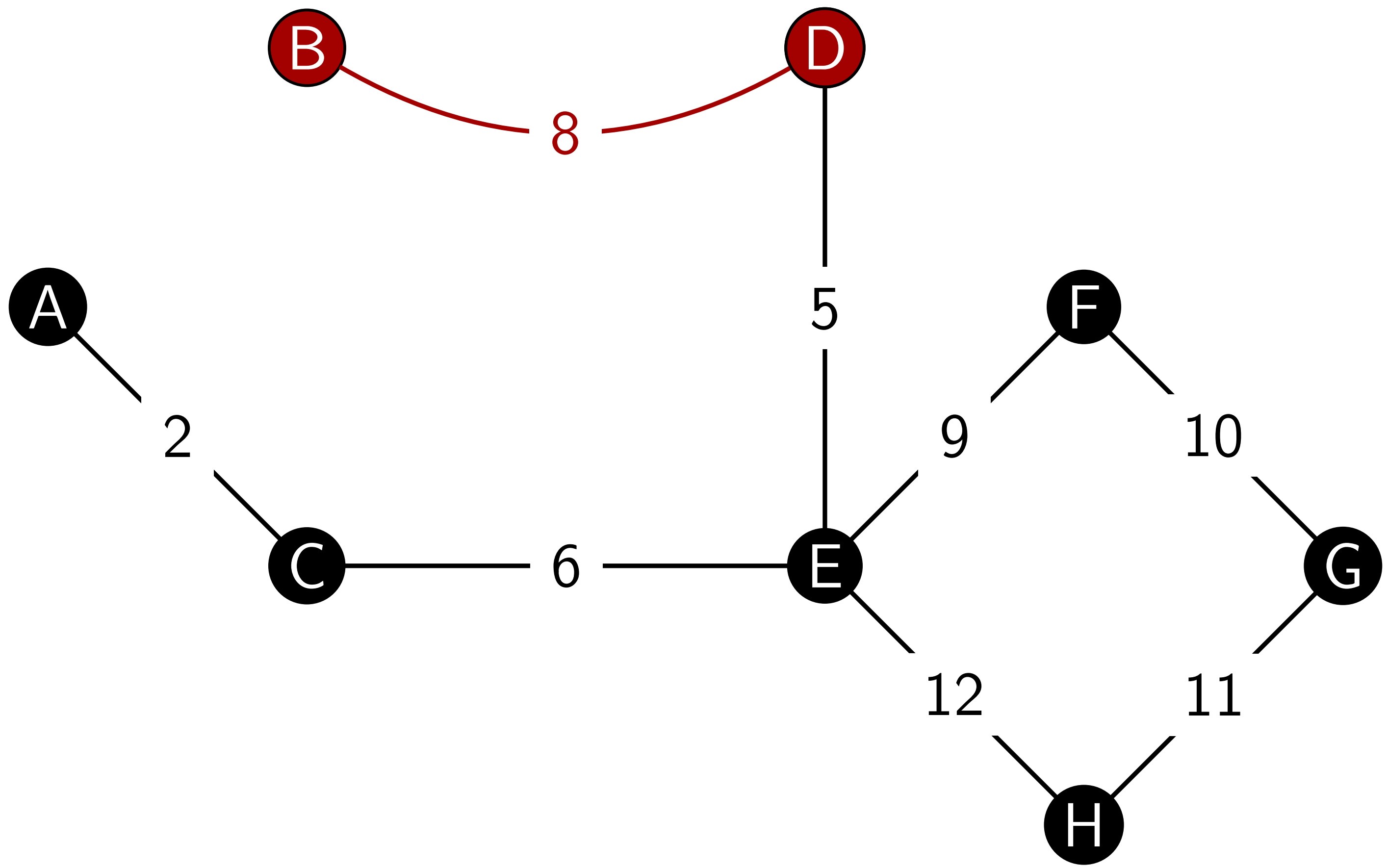
$E_E = 1, 3, 4$

Algorytm Fleury'ego



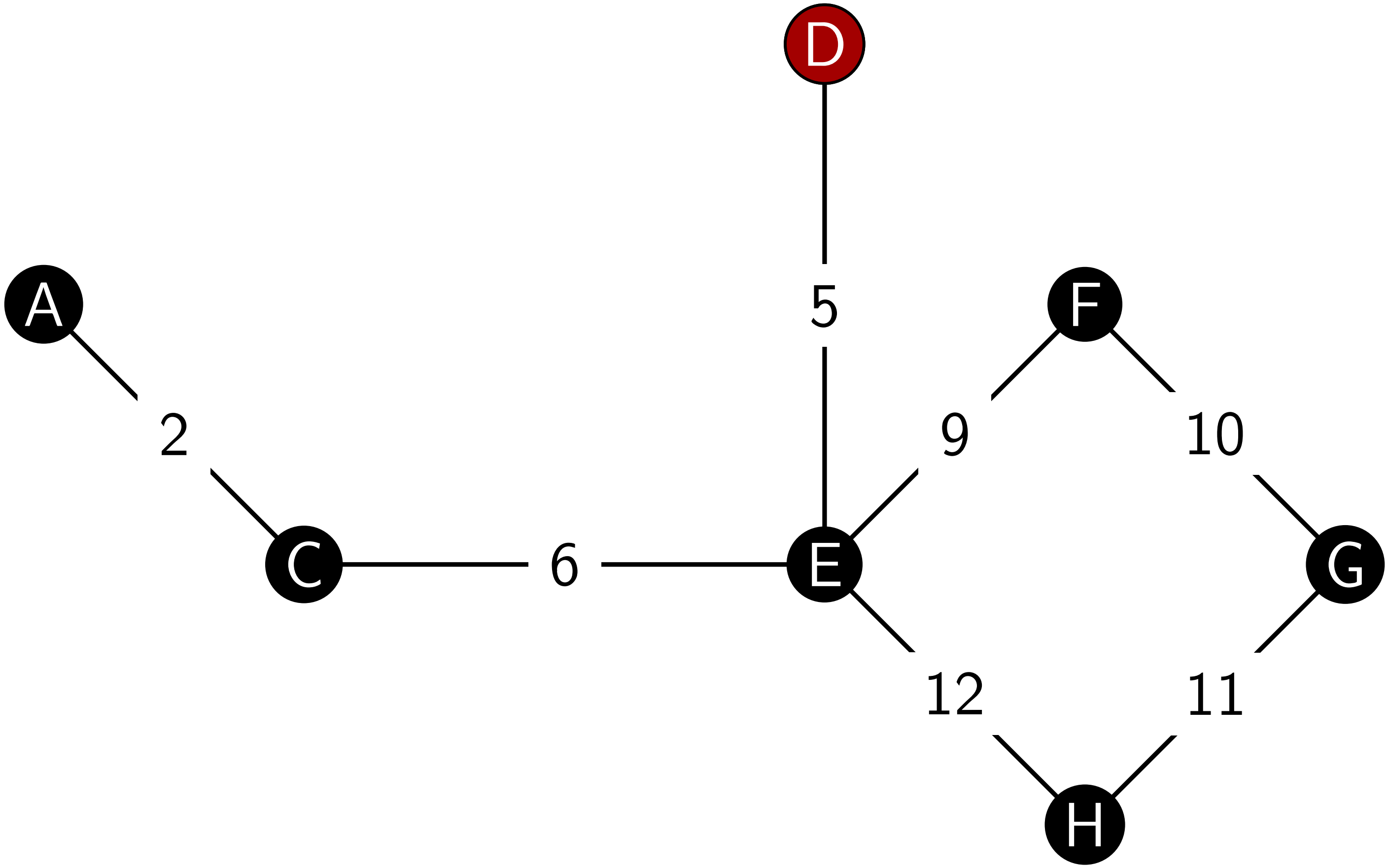
$E_E = 1, 3, 4, 7$

Algorytm Fleury'ego



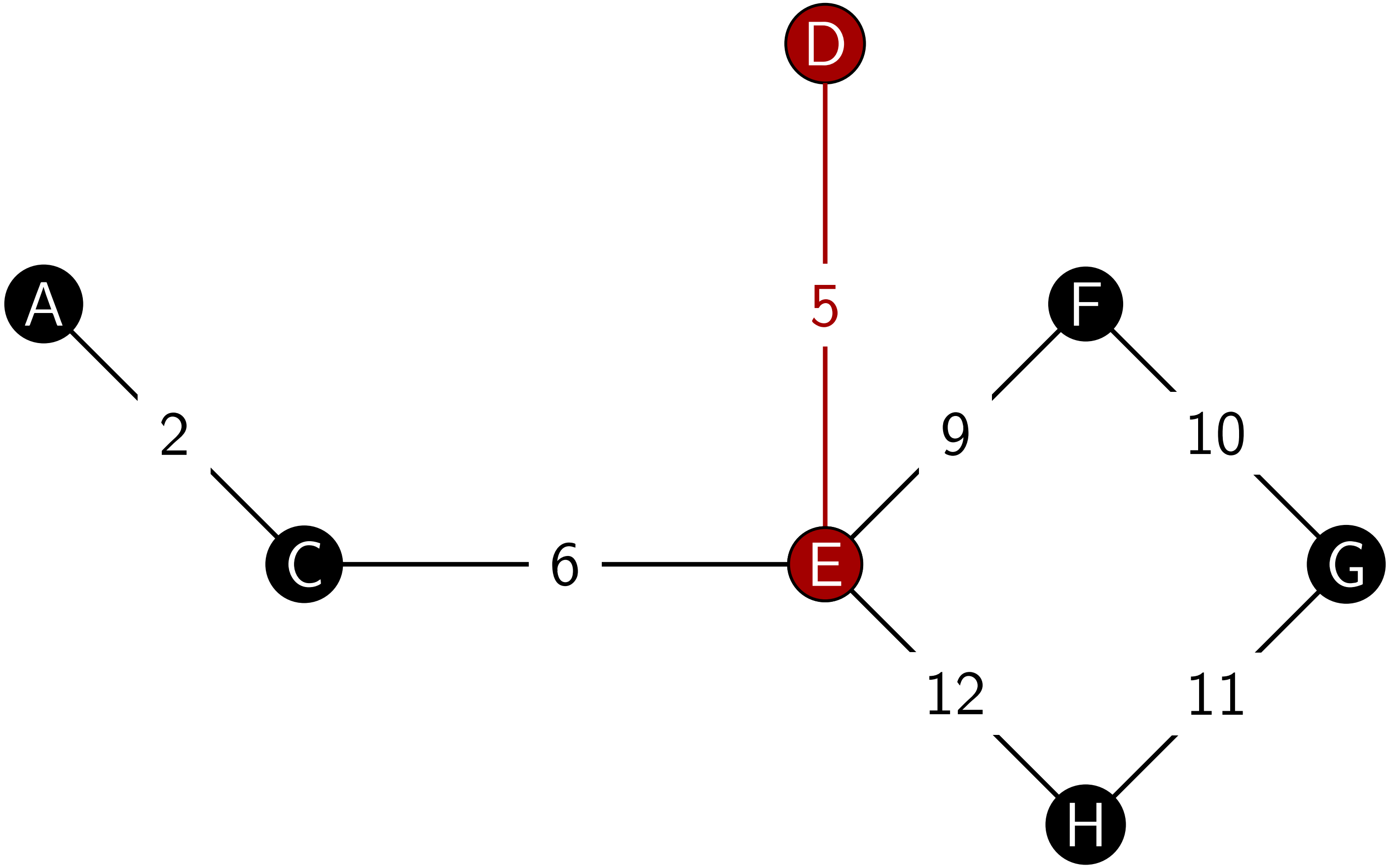
$E_E = 1, 3, 4, 7$

Algorytm Fleury'ego



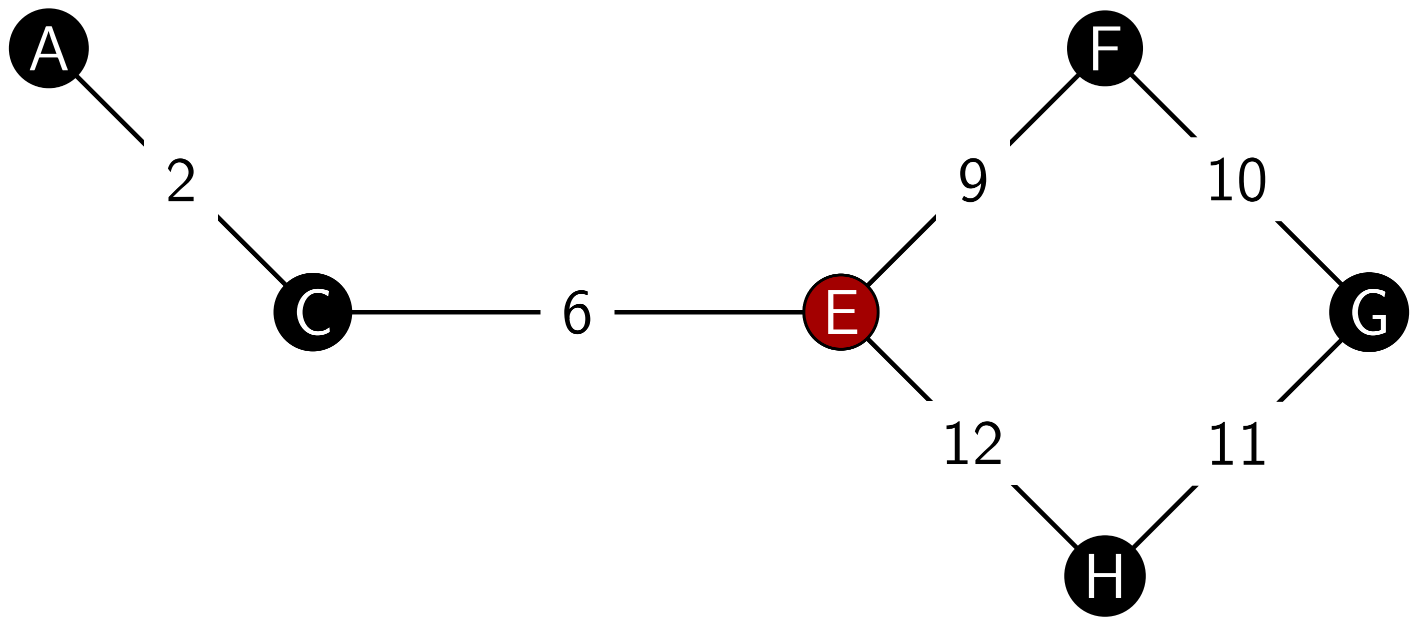
$E_E = 1, 3, 4, 7, 8$

Algorytm Fleury'ego



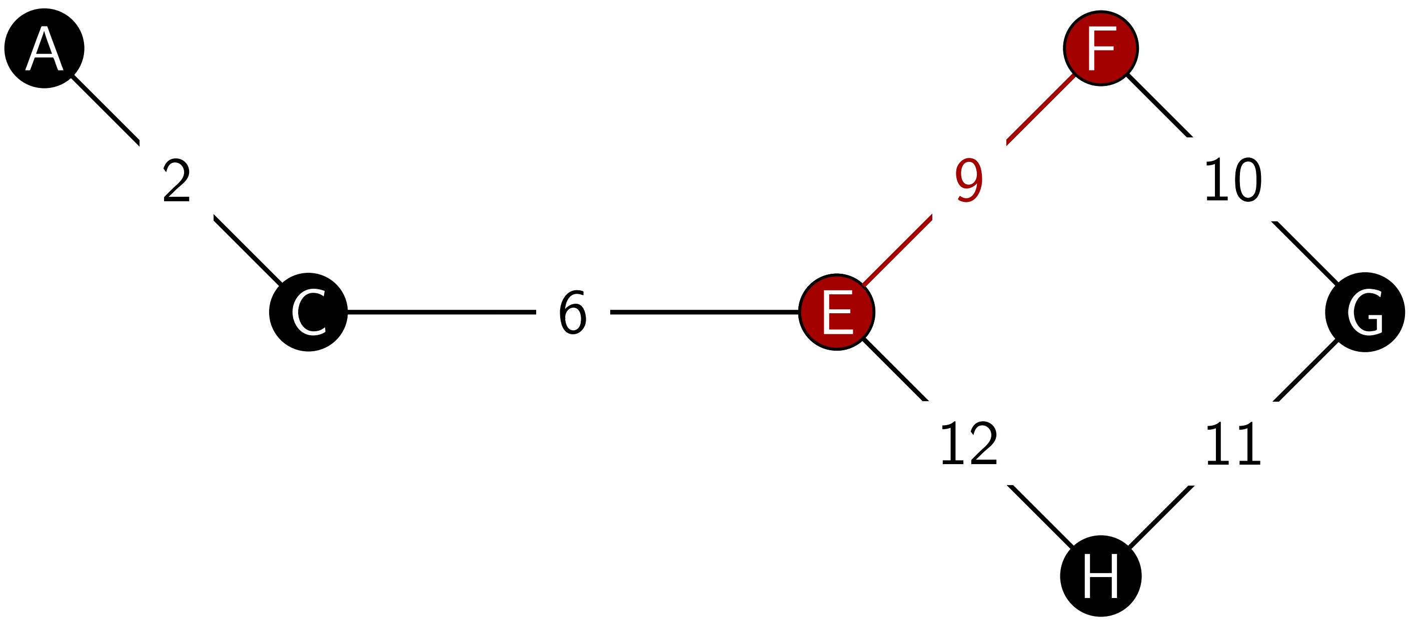
$E_E = 1, 3, 4, 7, 8$

Algorytm Fleury'ego



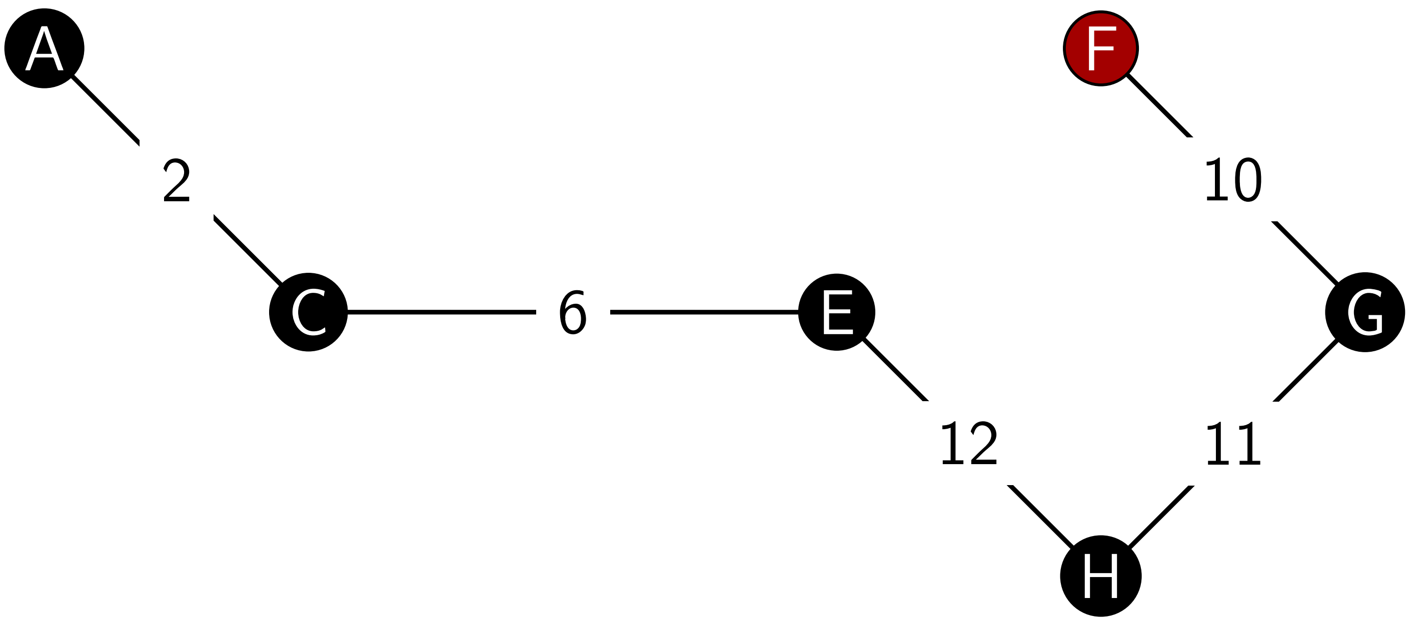
$E_E = 1, 3, 4, 7, 8, 5$

Algorytm Fleury'ego



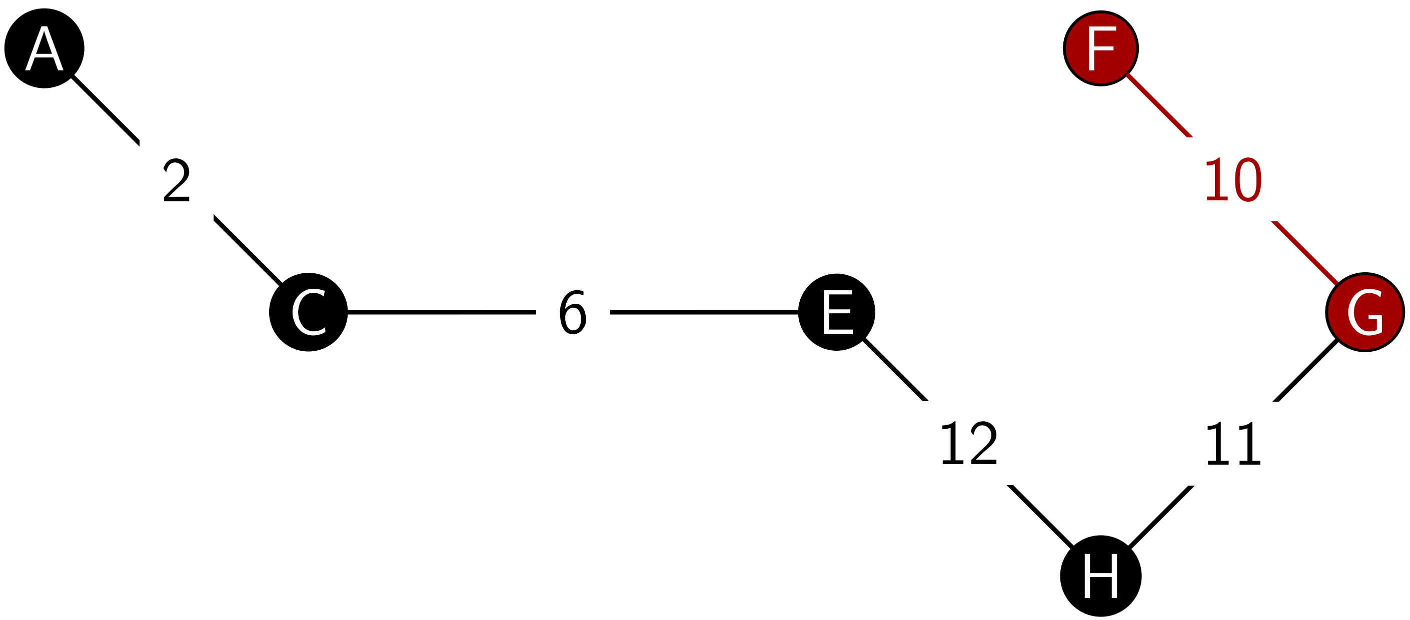
$$E_E = 1, 3, 4, 7, 8, 5$$

Algorytm Fleury'ego



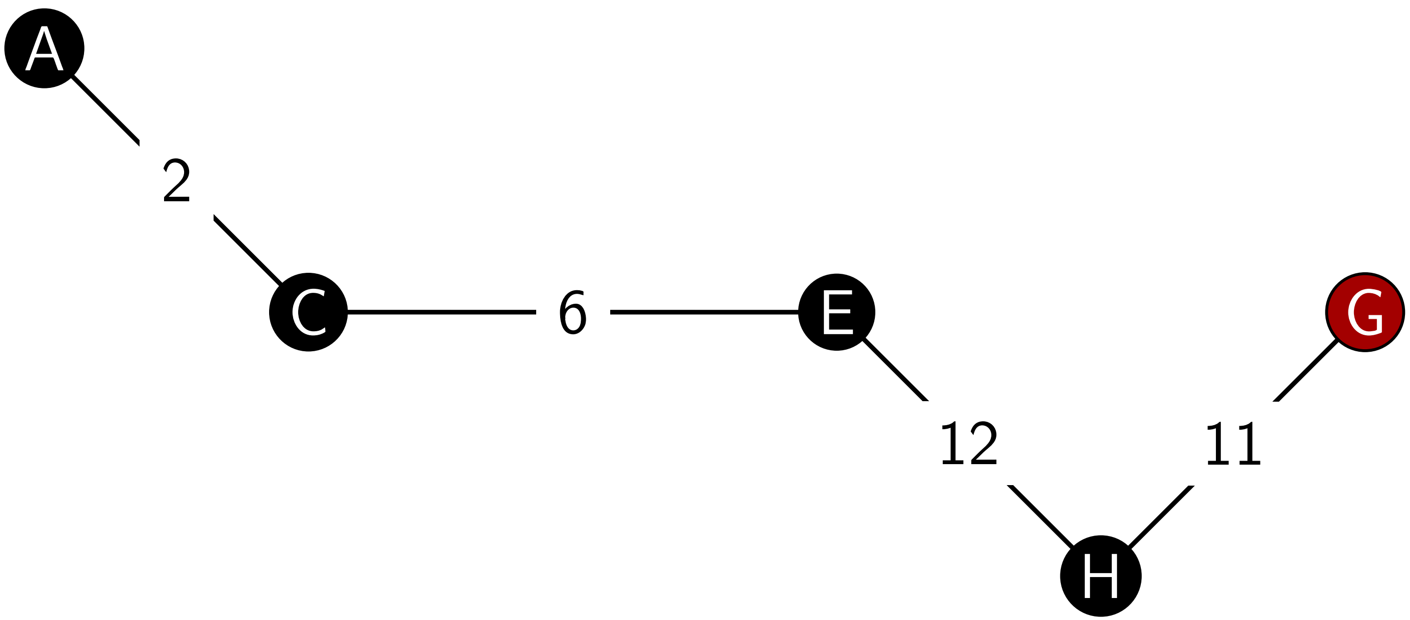
$E_E = 1, 3, 4, 7, 8, 5, 9$

Algorytm Fleury'ego



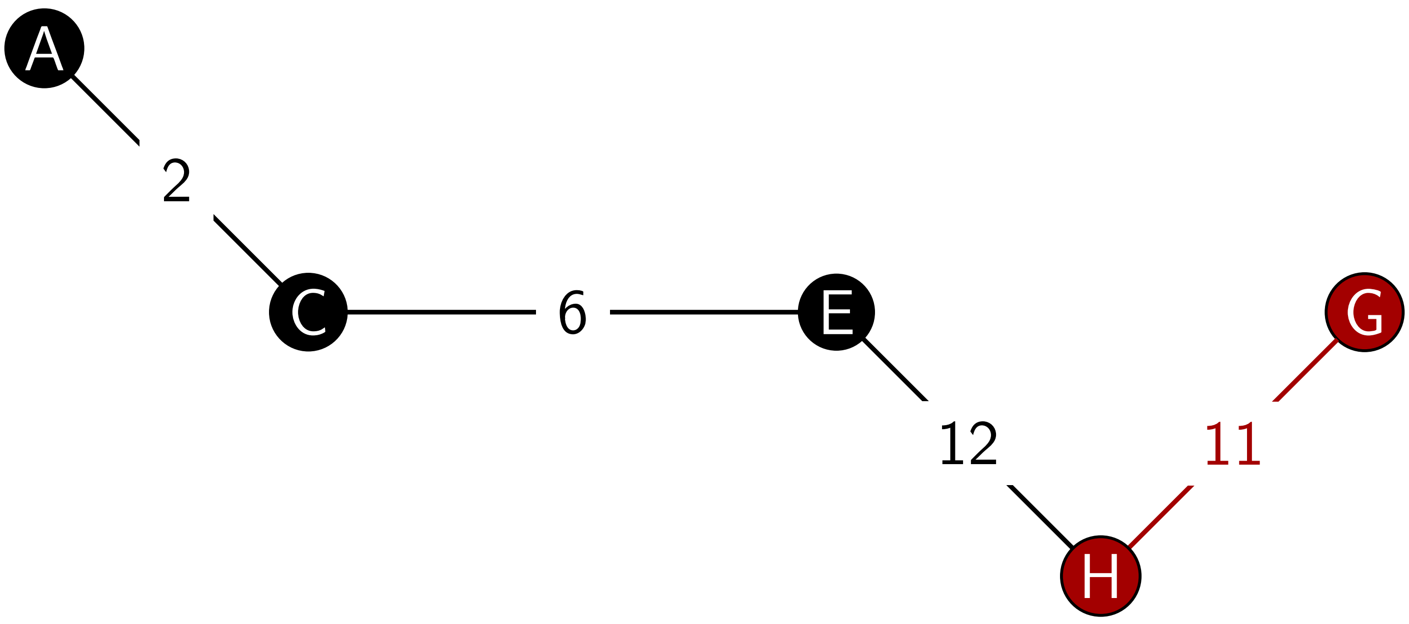
$E_E = 1, 3, 4, 7, 8, 5, 9$

Algorytm Fleury'ego



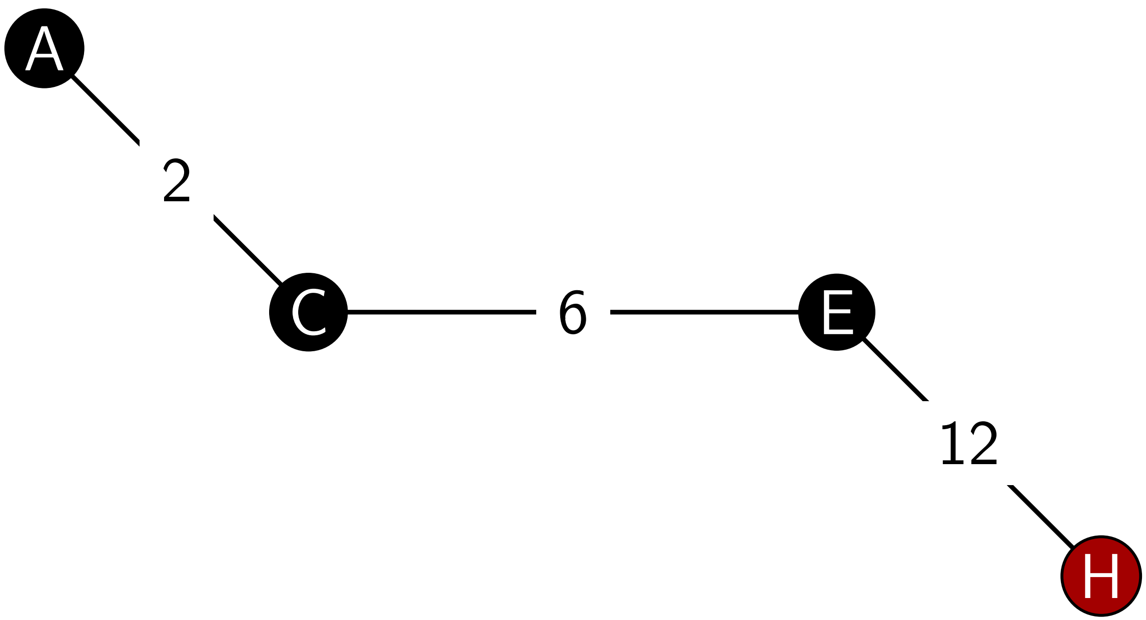
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10$$

Algorytm Fleury'ego



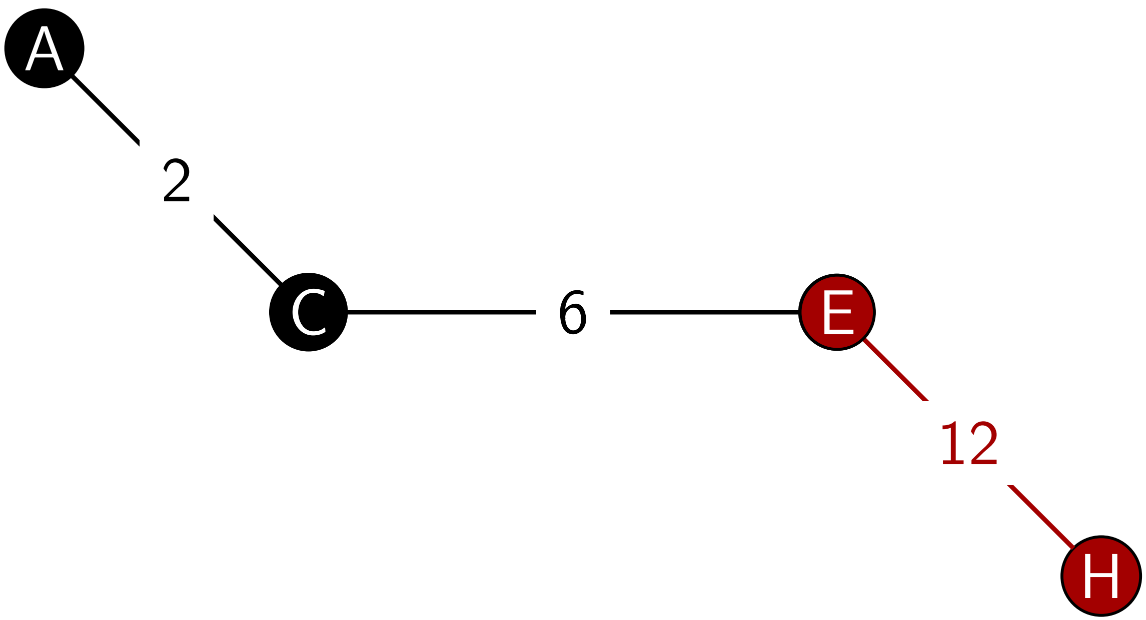
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10$$

Algorytm Fleury'ego



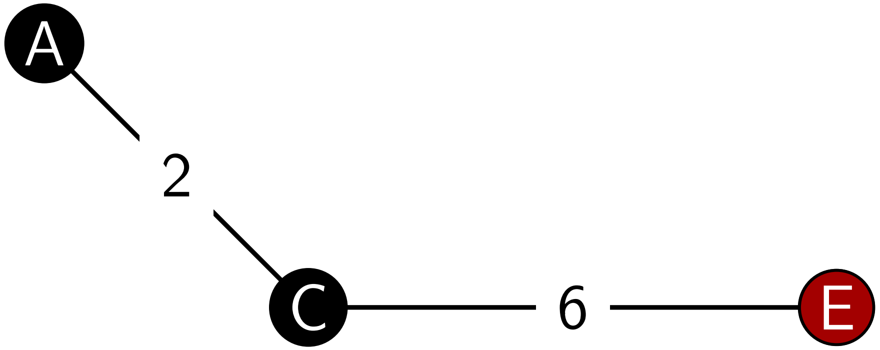
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11$$

Algorytm Fleury'ego



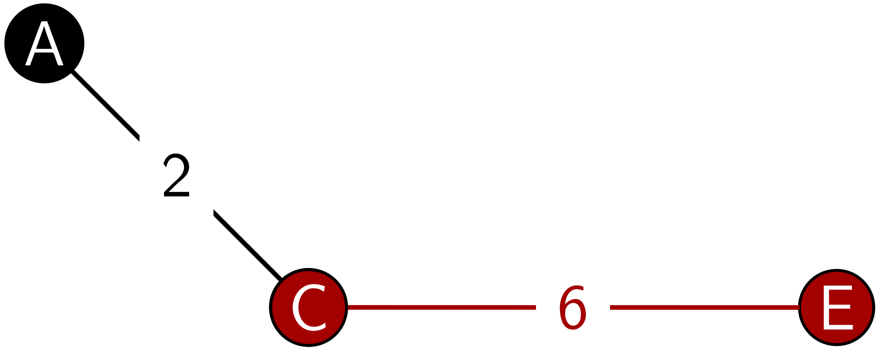
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11$$

Algorytm Fleury'ego



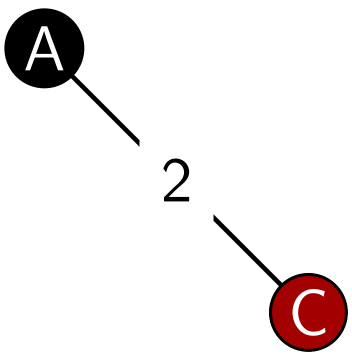
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11, 12$$

Algorytm Fleury'ego



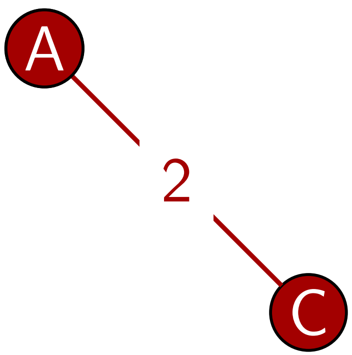
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11, 12$$

Algorytm Fleury'ego



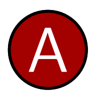
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11, 12, 6$$

Algorytm Fleury'ego



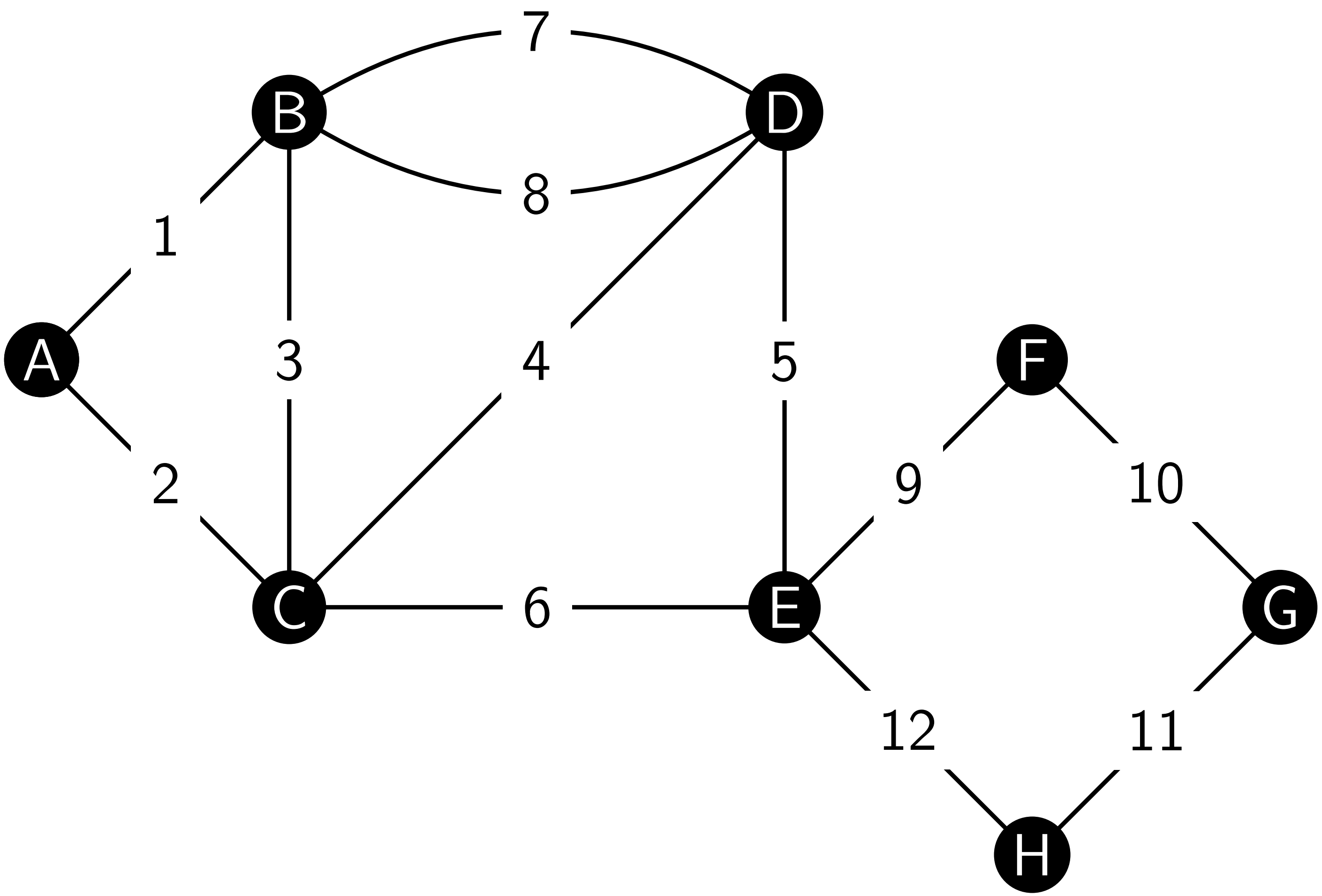
$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11, 12, 6$$

Algorytm Fleury'ego



$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11, 12, 6, 2$$

Algorytm Fleury'ego



$$E_E = 1, 3, 4, 7, 8, 5, 9, 10, 11, 12, 6, 2$$

Grafy hamiltonowskie

- **Cykl Hamiltona**: cykl zawierający wszystkie wierzchołki grafu.
- **Graf hamiltonowski**: graf zawierający cykl Hamiltona.

Grafy hamiltonowskie

- **Cykl Hamiltona**: cykl zawierający wszystkie wierzchołki grafu.
- **Graf hamiltonowski**: graf zawierający cykl Hamiltona.

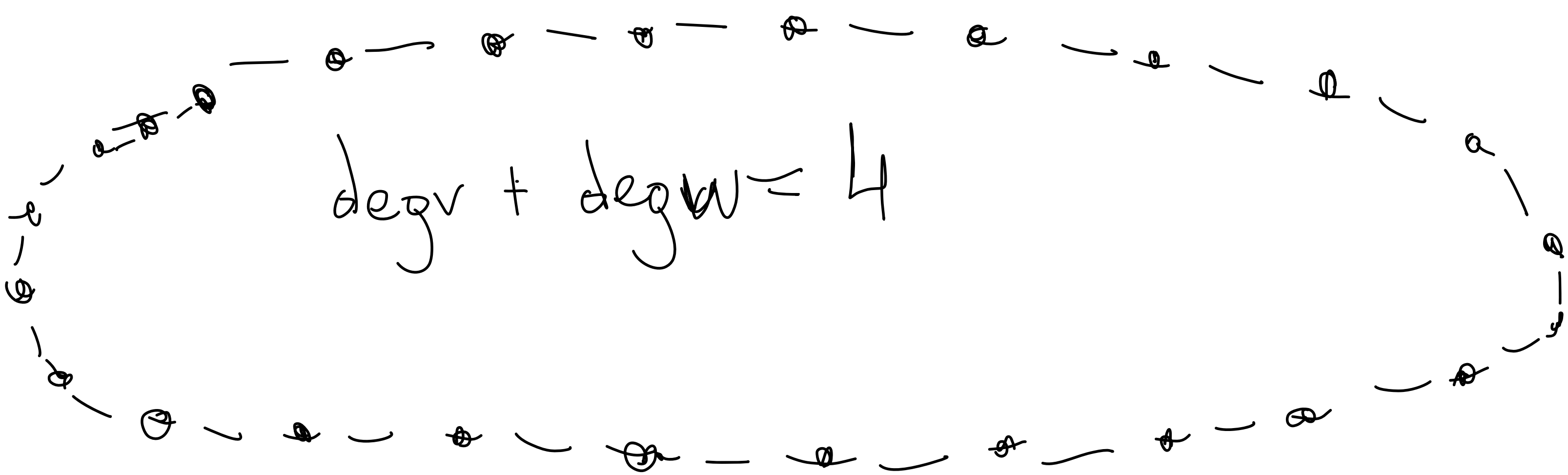
Twierdzenie Orego

Jeżeli graf $G = (V, E)$ ma przynajmniej trzy wierzchołki oraz dla każdej pary niesąsiednich wierzchołków v i w zachodzi

$$\boxed{\deg v + \deg w \geqslant |V|,}$$

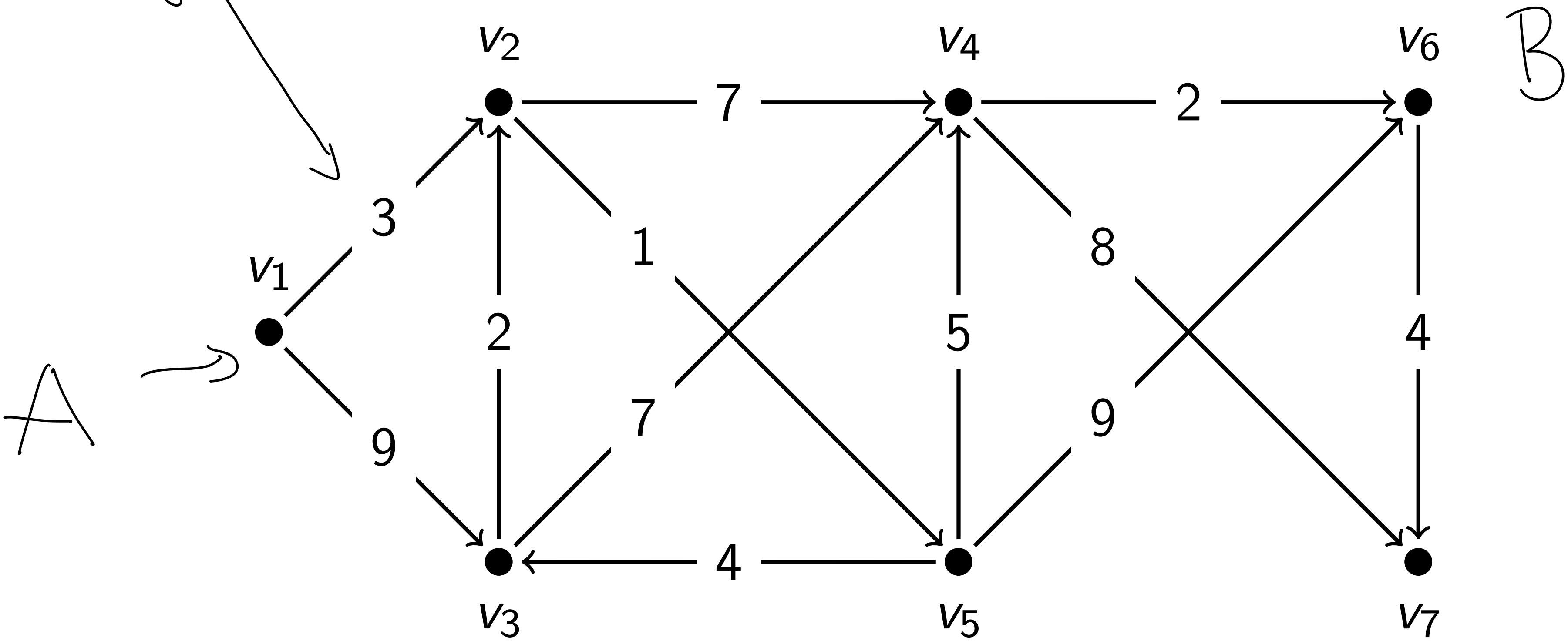
liczba wierzchołków

to graf G jest hamiltonowski.



Poszukiwanie ścieżek

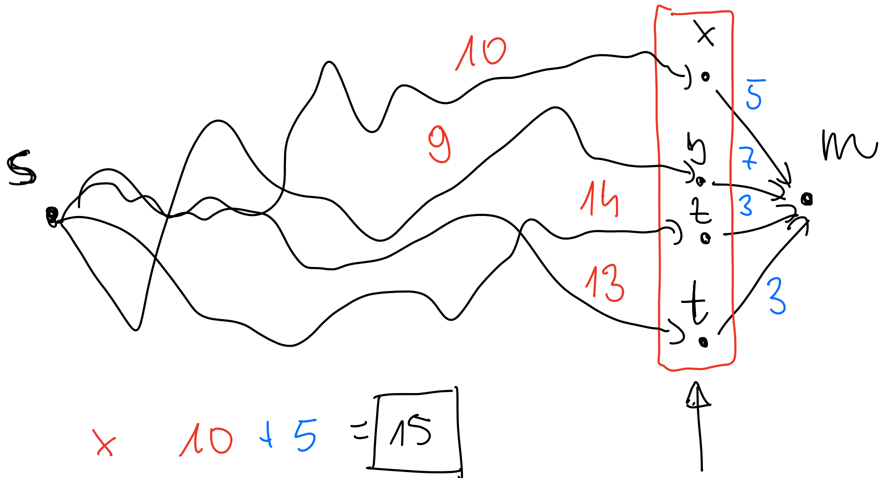
waga krawędzi



$w(i, j)$ = waga (koszt) krawędzi $v_i \to v_j$

$w(1,2) = 3$

Edgar Dijkstra (60')

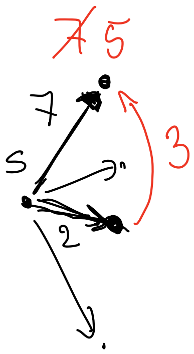


x $10 + 5 = \boxed{15}$

y $9 + 7 = 16$

z $14 + 3 = 17$

t $13 + 3 = 16$



Algorytm Dijkstry

input: graf skierowany $G = (\boxed{\{1, \ldots, n\}}, E)$, wagi $\boxed{W = W(v, w)}$

output: długość D_j najkrótszej ścieżki od 1 do j, $j \in \{2, \ldots, n\}$

$L \leftarrow \emptyset$ — wierzchołki, dla których znamy optymalne drogi

$V \leftarrow \{2, \ldots, n\}$

for $i \in V$ **do**

  $D_i \leftarrow W(1, i)$ — koszt dojścia w jednym kroku

end for

while $V \setminus L \neq \emptyset$ **do**

  wybierz $k \in V \setminus L$ o najmniejszym D_j

  $L \leftarrow L \cup \{k\}$

  for $j \in V \setminus L$ **do**

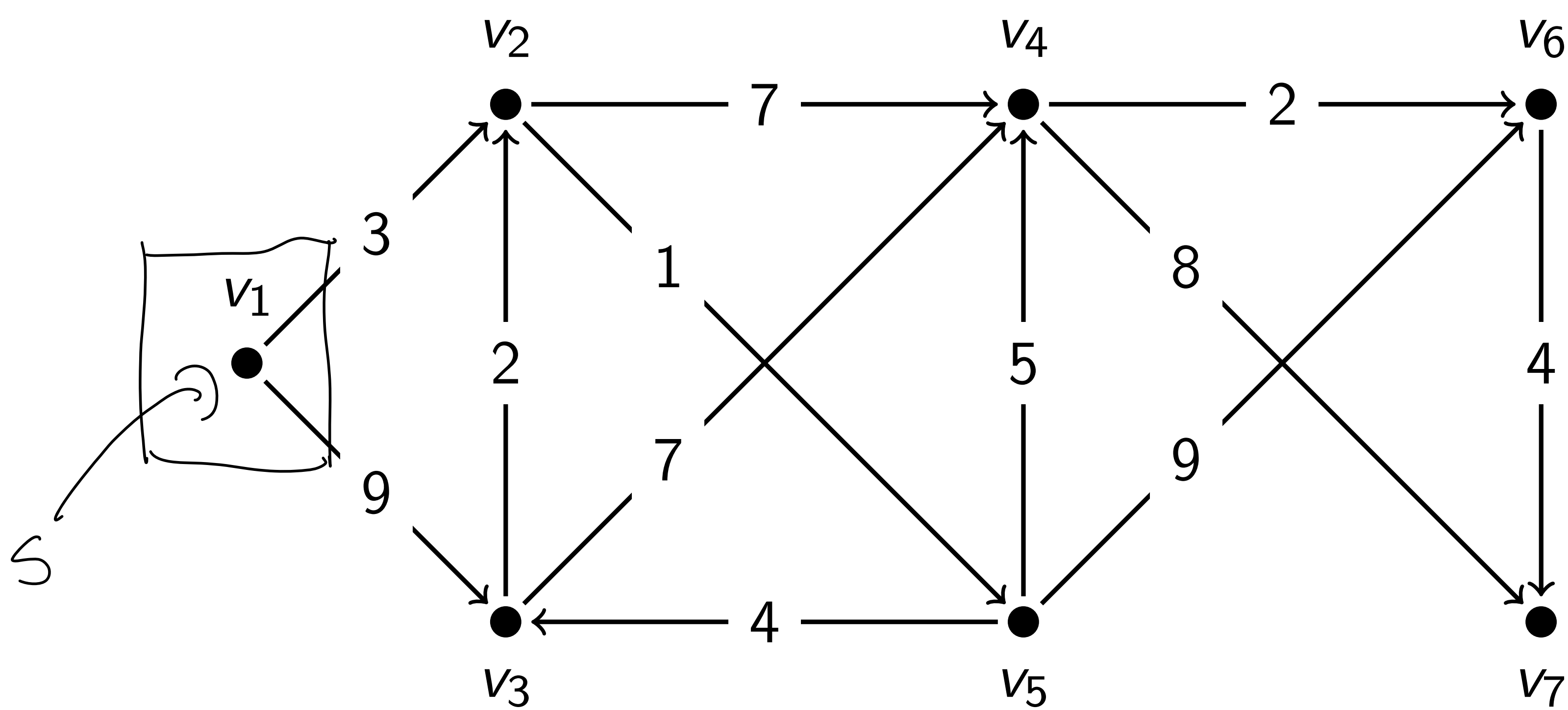
    if $D_j > D_k + W(k, j)$ **then**

      $D_j \leftarrow D_k + W(k, j)$

    end if

  end for

end while



L	D_2	D_3	D_4	D_5	D_6	D_7
	$\boxed{3}$	9	∞	∞	∞	∞
2		9	10	$\boxed{4}$	∞	∞
2,5		$\boxed{8}$	9		13	∞
2,5,3			$\boxed{9}$		13	∞
2,5,3,4					$\boxed{11}$	17
..., 6						$\boxed{15}$