Теория на графите. Дървета

Графи

- <u>Дефиниция:</u> **Граф** *G = (V, E)*, където:
  - $V \neq \emptyset$ е множество на възлите;
  - *E* - множество на връзките между възлите, наречени ребра.

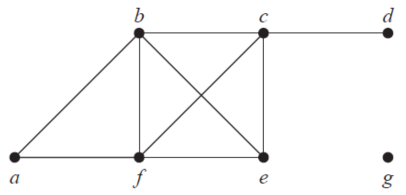
Ребрата са неподредена двойка от по два възела *e = {u, v}*.

*Казваме, че графите са **прости** или **ненасочени**, ако ребрата са им ненасочени, т.е. няма значение кой е първи и кой втори възел.*

- <u>Дефиниция:</u> Два възела *u* и *v* в ненасочения граф *G = (V, E)* се наричат ***съседни*** в *G*, ако *u* и *v* са крайни точки на реброто *e* в *G*. Реброто *e* се нарича ***инцидентно** с върховете u* и *v* или казваме, че *e свързва u* и *v*.
- <u>Дефиниция:</u> Броят на ребрата, инцидентни с върха $a \in V$ ще наричаме ***степен*** на върха a и ще означаваме с $\deg(a)$.
- <u>Дефиниция:</u> Един връх $a \in V$ се нарича ***изолиран***, ако $\deg(a) = 0$ и ***краен (лист),*** ако $\deg(a) = 1$.
- <u>Дефиниция:</u> За граф *G = (V, E)* с $V = \{v_1, \dots, v_n\}$ матрица на съседство наричаме квадратна матрица $M = (m_{i,j})_{1\le i,j\le n}$ с размерност броя на върховете $|V| = n$, където

$$m_{i,j} = \begin{cases} 1, & \text{ако } \{v_i, v_j\} \in E \\ 0, & \text{ако } \{v_i, v_j\} \notin E \end{cases}.$$

Пример: За ненасочения граф *G* имаме:



G

Степени:

$$\deg(a) = 2, \deg(b) = 4, \deg(c) = 4$$

$$\deg(d) = 1, \deg(e) = 3, \deg(f) = 4, \deg(g) = 0$$

Списък на съседство:	Матрица на съседство:
$a: b, f$ $b: a, c, e, f$ $c: b, d, e, f$ $d: c$ $e: b, c, f$ $f: a, b, c, e$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

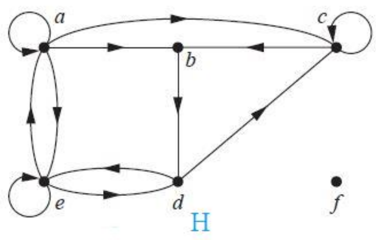
- <u>Дефиниция:</u> *G = (V, E)* е **насочен граф**, ако:
    - *G = (V, E)* е граф;
    - E - множество от наредени двойки върху *V*.

Ребрата в насочения граф се наричат клони.

*От ненасочен граф ще получим насочен, ако във всяко ребро дефинираме **начало** и **край**.*

- <u>Дефиниция:</u> Ако $a \in V$, то с $\deg^+(a)$ ще означаваме броя на ребрата с начало a (**изходящи** или **излизащи** ребра за a), а с $\deg^-(a)$ – броя на ребрата с край a (**входящи** или **влизащи** ребра за a).

Пример: За насочения граф *H* имаме:



Входящи:

$$\deg^-(a) = 2, \deg^-(b) = 2, \deg^-(c) = 3,$$
$$\deg^-(d) = 2, \deg^-(e) = 3, \deg^-(f) = 0$$

Изходящи:

$$\deg^+(a) = 4, \deg^+(b) = 1, \deg^+(c) = 2,\ \deg^+(d) = 2, \deg^+(e) = 3, \deg^+(f) = 0$$

Задачи:

<u>Задача 1.</u> Постройте ненасочените графи *G = (V, E):*

а) V = {*a, b, c, d, e*}
   E = {*aa, ac, bc, ad, de, ae*}

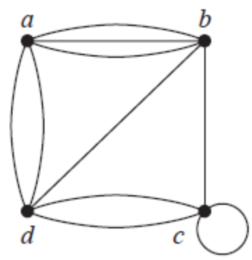
б) V = {*a, m, p, s, v*}
   E = {*am, ap, av, pv*}

<u>Задача 2.</u> За графите от ***задача 1***

а) постройте матрицата на съседство;

б) списъка на съседство.

<u>Задача 3.</u> За дадения граф:



а) постройте матрицата на съседство;

б) определете списъка на съседство;.

в) определете степените на върховете.

<u>Задача 4.</u> Постройте насочения граф *G = (V, E)*

V = {*a, b, c, d, e, f, g*}

E = *{(a ,a), (a, b), (b, a), (b, c), (c, e), (e, f), (f, d), (g, a)}*

а) постройте матрицата на съседство;

б) списъка на съседство;

в) определете степените на върховете.

Движения през граф

➢ <u>Дефиниция:</u> Нека $G = (V, E)$, е граф. ***Маршрут (walk) W*** с дължина $n > 0$ в G наричаме редицата:

$$v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n,$$

така че $v_k \in V$, а $e_k \in E$ за всяко $k = 1, \ldots, n$ като e_k свързва v_{k-1} и v_k, т.е. W свързва v_0 и v_n от v_0 към v_n.

- *Всяко ребро може да се разглежда като маршрут с дължина 1.*
- *Ако $v_0 = v_n$ и $n \geq 1$, то казваме че маршрута W е затворен. В противен случай е незатворен.*

- *Ако всички ребра в W са различни - се нарича* ***верига*** *(trail), (ако не е затворен).*
- *Ако всички възли му са различни се нарича* ***елементарен маршрут*** *или път.*
- *Ако W е затворен и е верига като всичките му възли са различни, казваме че W е* ***цикъл***.
- *Забелязваме, че в геометрична реализация елементарната верига образува проста незатворена линия, а елементарния цикъл - проста затворена линия.*

<u>Задача 5.</u> За графа от ***задача 1. а)*** проверете дали:

а) *b, c, a, d, a* е маршрут

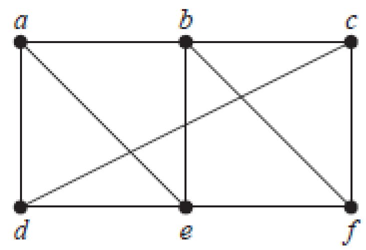
б) *b, c, a, d, a* е верига

в) *b, c, a, a, d, e* е елементарна верига

г) *a, a, d, e, a* е затворена верига

д) *a, a, d, e, a* е цикъл

<u>Задача 6.</u> За дадения граф определете вида на :



а) *a, d, c, f, e*

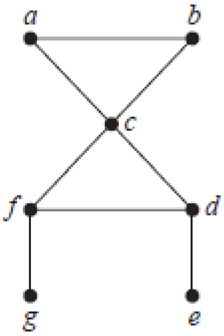
б) *d, e, c, a*

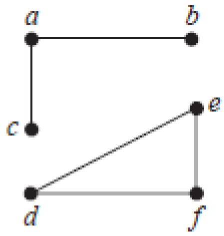
в) *b, c, f, e, b*

г) *a, b, e, d, a, b*

➢ <u>Дефиниция:</u> Казваме, че $G = (V, E)$ е ***свързан граф***, ако за всеки два възела съществува маршрут от единия към втория.

Задача 7. Определете дали следните графи са свързани или не

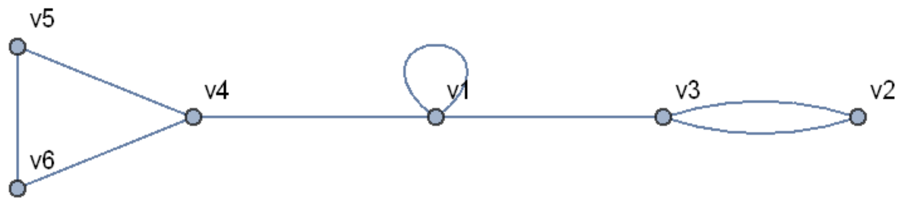




➢ ***Компонент*** на *G* е един максимален свързан подграф на *G*, (т.е. един свързан подграф на *G*), който не е подграф на никой друг свързан подграф на *G*.

Задача 8. Свързани ли са графите от ***задача 1***. Определете компонентите им, ако не са.

Задача 9. За графа:



а) Определете колко вериги можем да построим между *v1* и *v2* ?

б) Коя от тях е с най-малка дължина *r(v1, v2)*?

в) Коя е най-дългата верига, която можем да построим в графа?

г) А най-дългия път?

д) Има ли цикли – кои са те?

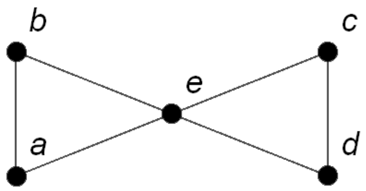
е) А паралелни ребра?

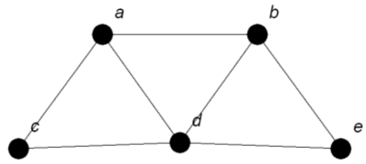
Ойлеров и Хамилтънов граф

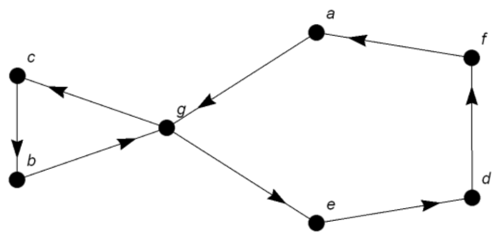
➢ Дефиниция: ***Цикъл (или тур) на Ойлер*** в G е затворена верига, която съдържа всяко ребро само веднъж. Всеки граф, който притежава ***Ойлеров цикъл*** се нарича ***Ойлеров граф***.

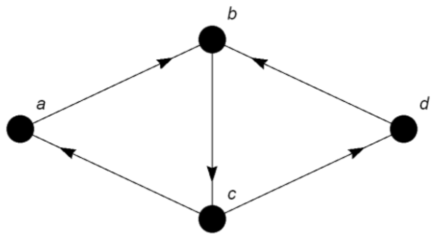
➢ Дефиниция: ***Ойлеров път*** в G е елементарна верига, която съдържа всяко ребро на G.

Задача 10. Определете дали следните насочени и ненасочени графи имат Ойлеров цикъл или Ойлеров път:





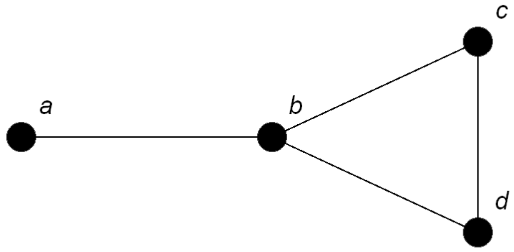


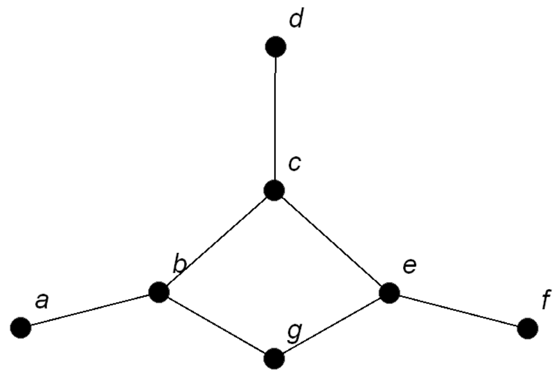


➢ <u>Дефиниция:</u> ***Цикъл на Хамилтън*** в G е цикъл, който съдържа всеки възел на графа. Граф, притежаващ Хамилтънов цикъл се нарича ***Хамилтънов граф.***

- ***Хамилтънов цикъл*** *в G е свързан подграф на G, съдържащ всичките му възли, в който всеки възел има степен 2.*
- *Всяка проста верига, съдържаща всички възли на един граф G е* ***Хамилтънова верига****.*
- *Ако един граф е* ***Хамилтънова верига****, но не е Хамилтънов цикъл, то той е* ***полухамилтънов*** *граф.*

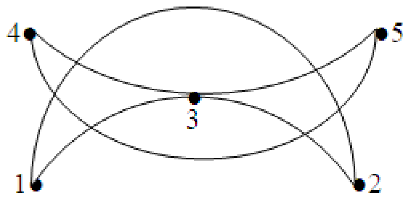
Задача 11. Определете дали следните графи имат Хамилтънов цикъл или Хамилтънова верига.

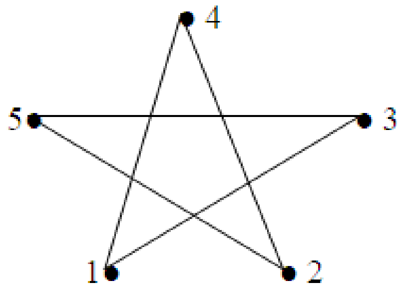




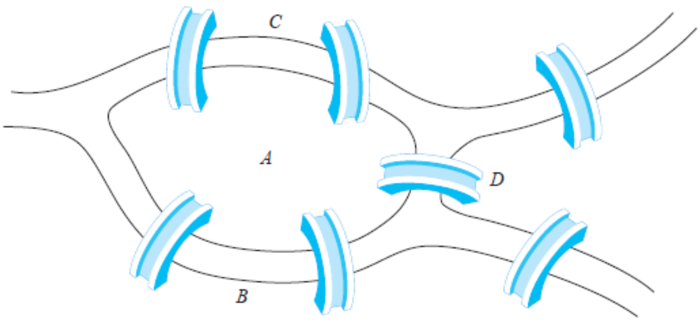
Задача 12. Покажете, че можете да начертаете следните графи без да вдигате молива. Определете дали са Ойлерови цикли, Хамилтънови цикли или Хамилтънови вериги:

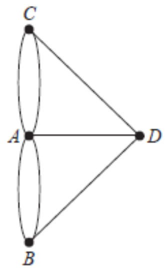
а)” Мохамедовите саби” б)





Задача 13. *Задачата за Кьонингсбергските мостове* Можем ли да обходим всички мостове като минаваме по тях само веднъж?

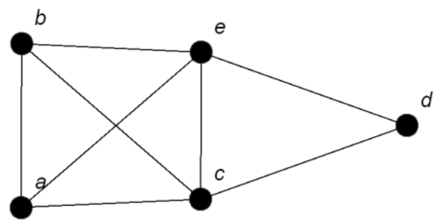




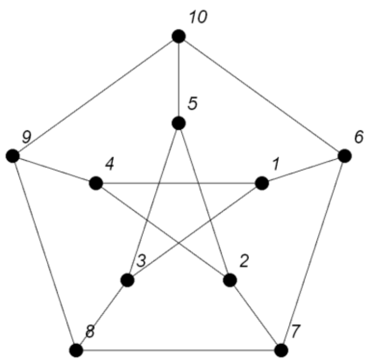
Задача 14. За дадения граф проверете дали е

а) Ойлеров цикъл

б) Хамилтънов цикъл



Задача 15. Граф на Петерсон



а) Може ли да построим Ойлеров цикъл?

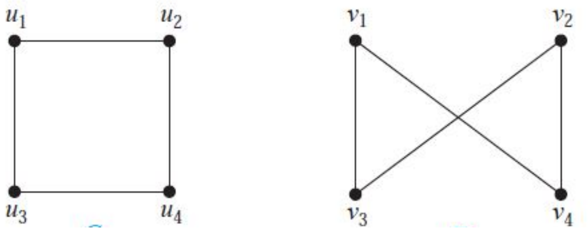
б) А Хамилтънов цикъл?

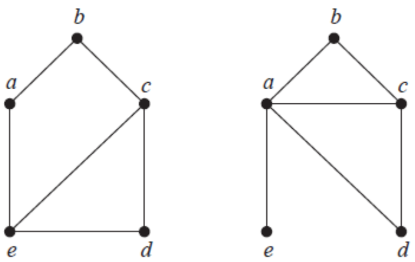
в) А Хамилтънова верига?

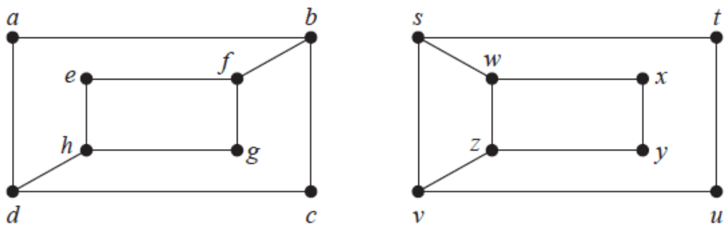
Изоморфизъм

➢ <u>Дефиниция:</u> Нека *G = (V, E)* и *H = (W, F)* са графи. Казваме, че са ***изоморфни***, ако съществува биекция $\varphi : V \rightarrow W$, така че за всяко *u*, *v* от V: $\{u,v\} \in E$ следва, че $\{\varphi(u), \varphi(v)\} \in F$. Тази биекция φ наричаме ***изоморфизъм*** между двата графа.

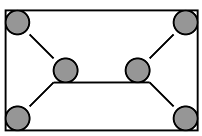
<u>Задача 16.</u> Определете дали следните двойки графите са изоморфни.

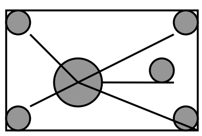
А) 

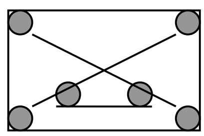
Б) 

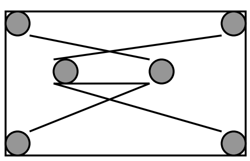
В) 

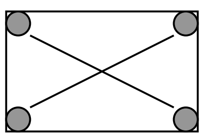
<u>Задача 17.</u> Някои от графите са изоморфни. Кои са те?

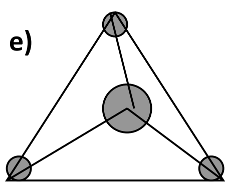
а) 

б) 

в) 

г) 

д) 

е) 

Дървета

➢ Дефиниция: ***Дървото*** е свързан, ориентиран граф, който не съдържа затворени вериги. Обикновено се бележи с Т.

- Дървото има един специален възел - корен.
- Дърво с корен бележим - (Т,r).
- Възли, от които не излиза ребро се наричат листа.
- Останалите - вътрешни възли.
- Дължината на пътя от корена до най-далечното листо се нарича височина на дървото.

➢ Дефиниция: Две дървета са ***изоморфни*** тогава и само тогава, когато съществува биекция между множествата на възлите им, която запазва съседите, не съседите и възела.

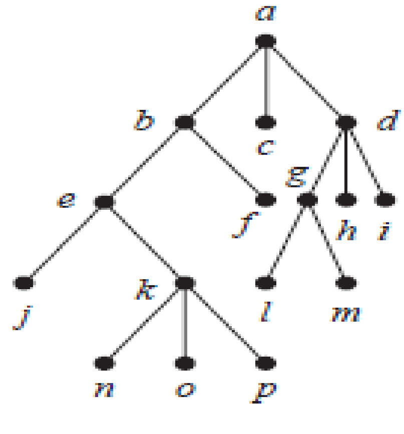
Обхождане на дървета

<table>
<tr><th>Алгоритъм за Preorder за генериране списък на възлите (от горе - надолу)</th><th rowspan="4">Алгоритъм за Inorder за генериране списък на възлите</th></tr>
<tr><td>procedure preorder (T: ordered rooted tree)
r:= root of T
list r
for each child c of r from left to right
T(c):=subtree with c as its root
preorder (T(c))</td></tr>
</table>

<table>
<tr><th>Алгоритъм за Inorder за генериране списък на възлите</th></tr>
<tr><td>procedure inorder (T: ordered rooted tree)
r:= root of T
if r is a leaf then list r
else
l:= first child of r from left to right
T(l):=subtree with l as its root
inorder (T(l))
list r
for each child c of r except for l from left to right
T(c):=subtree with c as its root
inorder (T(c))</td></tr>
</table>

<table>
<tr><th>Алгоритъм за Postorder за генериране списък на възлите (отдолу нагоре)</th></tr>
<tr><td>procedure postorder (T: ordered rooted tree)
r:= root of T
for each child c of r from left to right
T(c):=subtree with c as its root
postorder (T(c))
list r</td></tr>
</table>

Пример: Обхождане на дървото:

<table>
<tr><td rowspan="3">
</td><td>Preorder: обхождане първо на корена, а после на поддърветата от ляво на дясно
a b e j k n o p f c d g l m h i</td></tr>
<tr><td>Postorder: обхождане на поддърветата от ляво на дясно, а след това обхождане на корена
j n o p k e f b c l m g h i d a</td></tr>
<tr><td>Inorder
j e n k o p b f a c l g m d h i</td></tr>
</table>

Задачи:

<u>Задача 1.</u> Начертайте всички неизоморфни дървета с 5 възела.

<u>Задача 2.</u> Начертайте всички неизоморфни дървета с корен с 4 възела.

<u>Задача 3.</u> Начертайте пълно бинарно дърво с корен с

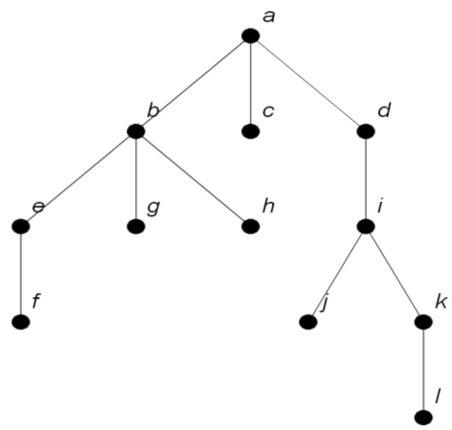
а) 11 възела б) 14 възела.

Възможно ли е? Колко листа и вътрешни възли има?

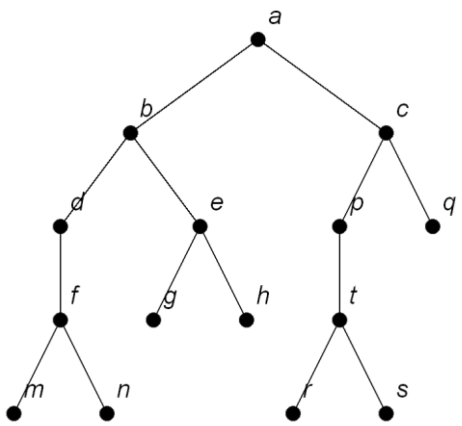
<u>Задача 4.</u> Как ще обходим дървото, ако използваме

а) преордер (от горе - надолу)

б) постордер (отдолу нагоре)



<u>Задача 5.</u> За двоичното дърво обходете възлите с **inorder**



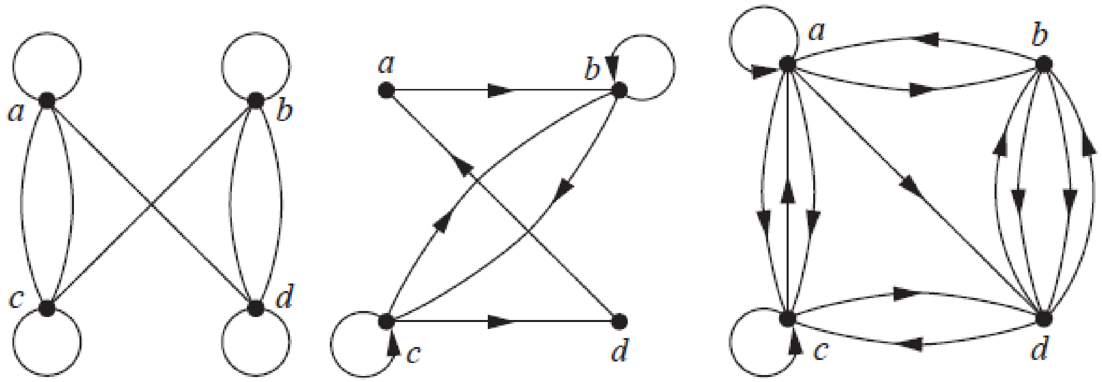
Допълнителни задачи:

Задача 1. За дадените графи:

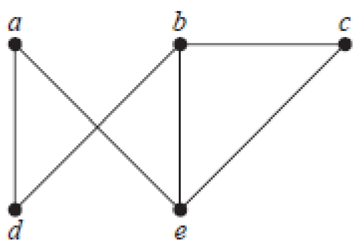
а) постройте матрицата на съседство;

б) определете списъка на съседство;.

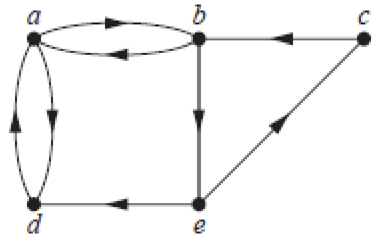
в) определете степените на върховете.



Задача 2. За дадените графи определете вида на:



а) *a, e, b, c, b*
б) *a, e, a, d, b, c, a*
в) *e, b, a, d, b, e*
г) *c, b, d, a, e, c*



а) *a, b, e, c, b*
б) *a, d, a, d, a*
в) *a, d, b, e, a*
г) *a, b, e, c, b, d, a*

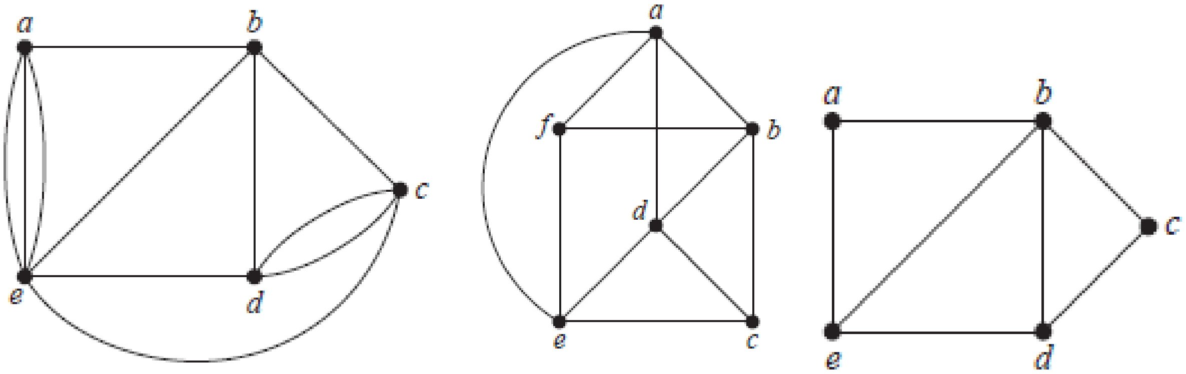
Задача 3. Определете дали в дадените графи има:

а) Ойлеров цикъл;

б) Ойлерови път;

в) Хамилтънов цикъл;

г) Хамилтънова верига.



Задача 4. Как ще обходим дървото, ако използваме метод за генериране на списък на възлите:

а) Preorder б) Postorder в) Inorder