1. Дефиниции за краен ориентиран (мулти)граф и краен неориентиран (мулти)граф.

- краен неориентиран граф

Определение 1: Граф, върхове и ребра

Граф е наредена двойка $G = (V, E)$, където V е непразно множество, чиито елементи се наричат *върхове*, E е множество, чиито елементи се наричат *ребра*, като

$$E \subseteq \{X \subseteq V : |X| = 2\}$$

Конвенция 1

Когато става дума за графи, буквата n означава броя на върховете, освен ако не е дефинирана иначе, и буквата m означава броя на ребрата, освен ако не е дефинирана иначе. Типичен запис е $V = \{v_1, v_2, \dots, v_n\}$ и $E = \{e_1, e_2, \dots, e_m\}$.

Конвенция 2

Друга полезна конвенция е, ако е дадено името на графа, да кажем G, но имената на множеството от върховете и на множеството от ребрата са неизвестни, да ги означаваме съответно с "$V(G)$" и "$E(G)$".

Определение 2: Празен и тривиален граф.

Нека $G = (V, E)$ е граф. Ако $E = \emptyset$, казваме, че G е *празен граф*. Ако $E \neq \emptyset$, G е *непразен*. Ако $|V| = 1$, казваме, че G е *тривиален граф*.

- краен неориентиран мултиграф

Определение 13: Мултиграф

Мултиграф е наредена тройка $G = (V, E, f_G)$, където V е непразно множество, чиито елементи се наричат *върхове*, E е множество, чиито елементи се наричат *ребра*, $V \cap E = \emptyset$ и

$$f_G : E \to \{X \subseteq V : |X| = 2\}$$

е *свързващата функция*.

- краен ориентиран граф

Определение 79: Ориентиран граф

Ориентиран граф е наредена двойка $G = (V, E)$, където V е непразно множество, чиито елементи се наричат *върхове*, E е множество, чиито елементи се наричат *ребра*, като

$$E \subseteq (V \times V) \setminus \{(u, u) \mid u \in V\}$$

Конвенция 10

Ако кажем само "граф", разбираме неориентиран граф. За да кажем, че имаме предвид ориентиран граф, трябва да кажем експлицитно "ориентиран".

- краен ориентиран мултиграф

Определение 86: Ориентиран мултиграф

Ориентиран мултиграф е наредена тройка $G = (V, E, f_G)$, където V е непразно множество, чиито елементи се наричат *върхове*, E е множество, чиито елементи се наричат *ребра*, $V \cap E = \varnothing$ и

$$f_G : E \to V \times V$$

е *свързващата функция*.

2. Дефиниции за път (цикъл) в ориентиран и неориентиран мултиграф.

Определение 16: Път.

Нека $G = (V, E)$ е граф. *Път в* G наричаме всяка алтернираща редица от върхове и ребра, за някое $t \geqslant 0$:

$$p = (u_{i_0}, e_{k_0}, u_{i_1}, e_{k_1}, u_{i_2}, \ldots, u_{i_{t-1}}, e_{k_{t-1}}, u_{i_t})$$

където $u_{i_p} \in V$ за $0 \leqslant p \leqslant t$, $e_{k_p} \in E$ за $0 \leqslant p \leqslant t-1$, и освен това е изпълнено $e_{k_p} = (u_{i_p}, u_{i_{p+1}})$ за $0 \leqslant p \leqslant t-1$. Върховете u_{i_0} и u_{i_t} се наричат *краищата на пътя*. Останалите върхове са *вътрешните върхове на пътя*. Още казваме, че p е път *между* u_{i_0} и u_{i_t} и в общия случай нямаме право да разменим u_{i_0} с u_{i_t}.
Дължината на пътя е броят на ребрата в него. Ще бележим дължината на пътя с $|p|$. В случая, $|p| = t$.
Ако всички елементи на пътя—върхове и ребра—са уникални, казваме, че p е *прост път*.

Определение 18: Цикъл.

Нека $G = (V, E)$ е граф и p е път в него, където:

$$p = (u_{i_0}, e_{k_0}, u_{i_1}, e_{k_1}, u_{i_2}, \ldots, u_{i_{t-1}}, e_{k_{t-1}}, u_{i_t})$$

Казваме, че p е *цикъл*, ако $u_{i_0} = u_{i_t}$. Казваме, че p е *прост цикъл*, ако p е цикъл с поне едно ребро и освен това, всички елементи освен $u_{i_0} = u_{i_t}$ са уникални.

3. Свързаност и свързани компоненти на граф.

Определение 19: Свързаност в граф. Свързан граф.

Нека $G = (V, E)$ е граф. За всеки два върха $u, v \in V$ казваме, че u и v са *свързани*, ако съществува u-v път. G е *свързан*, ако всеки два върха в него са свързани.

Определение 20: Свързани компоненти.

Нека $G = (V, E)$ е граф и Q_G е релацията на достижимост върху G. Подграфите на G, индуцирани от класовете на еквивалентност на Q_G, се наричат *свързаните компоненти на* G.

Определение 21: Свързани компоненти, алтернативно определение.

Свързаните компоненти на граф са максималните по включване свързани подграфи.

4. Дефиниция на дърво и кореново дърво.

Определение 43: Дърво, не-индуктивна дефиниция.

Дърво е всеки граф, който е свързан и ацикличен.

Определение 44: Дърво, индуктивна дефиниция.

Множеството от дърветата се дефинира така:

❶ **База** Всеки тривиален граф е дърво.

❷ **Индуктивна стъпка** Ако $T = (V, E)$ е дърво и u е връх в T и w е връх, който не е в T, то $T' = (V \cup \{w\}, E \cup \{(u, w)\})$ е дърво.

Лема 9

Граф е дърво тогава и само тогава, когато между всеки два негови върха има точно един път.

Определение 47: Кореново дърво, глобална дефиниция. Родител и дете.

Нека $T = (V, E)$ е дърво. Избираме произволен връх $r \in V$ и го наричаме *корен*. След избора на корен T става *кореново дърво*. Изборът на корен еднозначно определя релация на *родителство* върху всяко ребро. Нека $e \in E$ е произволно ребро и нека $e = (u, v)$. Съгласно Лема 9, съществува точно един път p между r и u и съществува точно един път q между r и v. Нещо повече.

- Или $V(p) = V(q) \cup \{u\}$, в който случай v е предпоследният връх преди u в p и казваме, че v *е родителят на* u, а u *е дете на* v:

p


- или $V(q) = V(p) \cup \{v\}$, в който случай u е предпоследният връх преди v в q и казваме, че u *е родителят на* v, а v *е дете на* u.

q


Определение 49: Предшественици и наследници на връх.

Нека T е кореново дърво с корен r и u е произволен връх в него. За всеки връх v казваме, че v *е предшественик на* u, ако v е връх от u-r пътя. За всеки връх v казваме, че v *е наследник на* u, ако u е предшественик на v.

Определение 55: Кореново дърво, първа индуктивна дефиниция.

❶ **База** Всеки тривиален граф $(\{u\}, \varnothing)$ е кореново дърво с корен u, множество от листа $\{u\}$, разклоненост 0 и височина 0.

❷ **Индуктивна стъпка** Нека $T_1 = (V_1, E_1)$, …, $T_k = (V_k, E_k)$ са коренови дървета, които две по две нямат общи върхове, с корени съответно $r_1, \ldots, r_k$, множества от листа съответно W_1, …, W_k, разклонености съответно b_1, …, b_k и височини съответно h_1, …, h_k. Нека r е връх, който не се намира в никое от тях. Нека $E' = \{(r, r_i) \,|\, 1 \leqslant i \leqslant k\}$. Тогава

$$T = \left(\{r\} \cup \bigcup_{i=1}^{k} V_i, E' \cup \bigcup_{i=1}^{k} E_i\right)$$

е кореново дърво с корен r, множество от листа $\bigcup_{i=1}^{k} W_i$, разклоненост $\max\{k, b_1, \ldots, b_k\}$ и височина $\max\{h_1, \ldots, h_k\} + 1$.

Определение 56: Кореново дърво, втора индуктивна дефиниция.

❶ **База** Всеки тривиален граф $(\{u\}, \varnothing)$ е кореново дърво с корен u, множество от листа $\{u\}$, разклоненост 0 и височина 0.

❷ **Индуктивна стъпка** Нека $T_1 = (V_1, E_1)$, и $T_2 = (V_2, E_2)$ са дървета, които нямат общи върхове, с корени съответно r_1 и r_2, множества от листа съответно W_1 и W_2, разклонености съответно b_1 и b_2 и височини съответно h_1 и h_2. Нека $v \in V_1$, нека v има f_v деца и нека дълбочината на v е t_v. Тогава

$$T = (V_1 \cup V_2, E_1 \cup E_2 \cup \{(v, r_2)\})$$

е кореново дърво с корен r_1. Множеството от листата на T е $(W_1 \setminus \{v\}) \cup W_2$, разклонеността му е $\max\{b_1, b_2, f_v + 1\}$, а височината му е $\max\{h_1, t_v + 1 + h_2\}$.

Определение 57: Кореново дърво, трета индуктивна дефиниция.

❶ **База** Всеки тривиален граф $(\{u\}, \varnothing)$ е кореново дърво с корен u, множество от листа $\{u\}$, разклоненост 0 и височина 0.

❷ **Индуктивна стъпка** Нека $T = (V_T, E_T)$, е коренови дървета с корен съответно r, множества от листа W, разклоненост b и височини съответно h. Нека z_1, …, z_q са върхове, които не са във V_T, и нека $\ell \in W$. Тогава

$$D = (V_T \cup \{z_1, \ldots, z_q\}, E_T \cup \{(\ell, z_1), \ldots, (\ell, z_q)\})$$

е кореново дърво с корен r и множество от листа $(W \setminus \{\ell\}) \cup \{z_1, \ldots, z_q\}$. Разклонеността на D е $\max\{b, q\}$. Височината на D е или h, ако дълбочината на ℓ в T е по-малка от h, или $h + 1$, ако ℓ в T е h.

5. Доказателство, че всяко кореново дърво е дърво и |V|=|E|+1.

Лема 11

Във всяко дърво, $m = n - 1$.

Доказателство: Със структурна индукция. В базовия случай на Определение 44, очевидно за графа с един връх и нула ребра, твърдението е вярно. ✓

Нека твърдението е вярно за дървото T от индуктивната стъпка. С други думи, допускаме, че

$$|E(T)| = |V(T)| - 1 \tag{2.11}$$

Трябва да докажем, че твърдението е вярно за дървото T'. С други думи, да докажем, че

$$|E(T')| = |V(T')| - 1 \tag{2.12}$$

Но това е съвършено очевидно предвид факта, че $|E(T')| = |E(T)| + 1$ и $|V(T')| = |V(T)| + 1$; ако заместим $|E(T')|$ с $|E(T)| + 1$ и $|V(T')|$ с $|V(T)| + 1$ в (2.12), ще получим равенство, еквивалентно на (2.11).

Лема 8 казва, че дърво с поне два върха има поне два висящи върха. Лема 12 дава точния брой на висящите върхове, изразен чрез броя на върховете от степен поне 3. Забележете, че върховете от степен 2 нямат значение за броя на висящите върхове; в сумата бихме могли да сумираме и по тях, понеже, ако $d(v) = 2$, очевидно $d(v) - 2 = 0$, така че техният "принос" би бил нула.

6. Покриващо дърво на граф.

Определение 59: Покриващо дърво.

Нека $G = (V, E)$ е свързан граф. *Покриващо дърво* на G е всяко дърво $T = (V, E')$, където $E' \subseteq E$.

7. Обхождане на граф в ширина и дълбочина.

- **BFS**

Как работи BFS. Много общо казано, той започва от дадения стартов връх, обхожда неговите съседи, като при това обхожда ребрата, с които ги достига, после обхожда съседите на съседите (различни от стартовия връх), и така нататък, обхождайки върховете по нарастване на разстоянията от стартовия връх. Версията на BFS, която ще разгледаме, ползва не две, а три състояния за всеки връх i:

1. i е непосетен. Всички върхове в началото са непосетени. Условно казваме, че непосетените върхове са *бели*.

2. i е посетен, но още не сме приключили с него. В метафората с лабиринта, такова е състоянието на стая, в която вече сме били, но все още не сме използвали всички врати, водещи навън от нея; с други думи, не сме обходили коридорите, излизащи от нея. Условно казваме, че върховете в това състояние са *сиви*.

3. i е посетен и вече сме приключили с него. В метафората с лабиринта, това е стая, в която сме били и освен това сме обходили всички коридори, излизащи от нея. Условно казваме, че върховете в това състояние са *черни*.

И така, BFS, който ще разгледаме, ползва масив $\mathrm{colour}[1, \ldots, n]$, всеки елемент от който има точно една от стойностите `white`, `grey` и `black`.

- DFS

DFS е алгоритъмът за обхождане на графи в дълбочина. Името идва от **D**epth-**F**irst **S**earch. За разлика от BFS, който е построен директно върху Схема 1 с реализация на множеството S чрез опашка, DFS не се получава от Схема 1 директно – факт, който обосноваваме на стр. 303. Въпреки че DFS не се получава директно от Схема 1, ние ще ползваме нейното доказателство за коректност и за него.

DFS наистина е свързан със стекова структура (LIFO), но имплементацията, която ще разгледаме, е рекурсивен алгоритъм, който ползва неявно системния стек.

Интуитивно казано, DFS обхожда графа по начин, обратен на BFS. BFS е "предпазливото" обхождане, при което обхождаме слой по слой навън от избрания начален връх. DFS е "смелото" обхождане, при което—ако ползваме аналогията с лабиринта—попадайки в коя да е стая,

- излизаме през първата неползвана врата, ако има такава, и отиваме в стая, в която или не сме били, или вече сме били:
  - ако не сме били, маркираме я като посетена и продължаваме по същия начин,
  - а ако сме били, се връщаме обратно
- ако вече няма неизползвана врата, се връщаме колкото е възможно по-малко и опитваме същото нещо, ако е възможно; когато сме отново в началната стая и вече няма неизползвани врати, прекратяваме алгоритъма.

8. Ойлерови обхождания на мултиграф.

?

9. Теореми за съществуване на Ойлеров цикъл (с доказателство) и Ойлеров път.

Определение 42: Ойлеров цикъл и Ойлеров път

Ойлеров цикъл в G е цикъл, не непременно прост, който съдържа всяко ребро точно веднъж. *Ойлеров път* в G е път, не непременно прост, който съдържа всяко ребро точно веднъж. G е *Ойлеров*, ако има Ойлеров цикъл.

Теорема 20: Ойлеров цикъл в свързан мултиграф

G има Ойлеров цикъл тогава и само тогава, когато всеки връх има четна степен.

Доказателство, I: Първо да допуснем, че G има Ойлеров цикъл c. Ще покажем, че G има върхове само от четна степен. Разглеждаме произволен $u \in V(G)$. Очевидно $u \in V(c)$.

Да си представим c като кръгова, а не линейна, наредба. Определение 18 говори за цикъла като за линейна наредба, в която първият и последният връх съвпадат, което съвпадане задава цикличността. Може да си мислим обаче за всеки цикъл с ненулева дължина като за истинска **кръгова наредба**, в която цикличността е естествена; тоест алтерниращата редица от върхове и ребра, по равен брой от всеки вид, е записана върху окръжност. Всяка поява на u в c отговаря на точно две ребра – това са съседните елементи на u в c.

Ако u няма примки, няма как два съседни върха в c (естествено, между тях в c има ребро) да са u. Следователно, на всяка поява на u в c съответстват точно две ребра—съседните елементи на u в цикъла—като за всеки две различни появи на u, двете двойки ребра нямат общ елемент. Следователно, множеството от всички тези двойки ребра, върху всички появи на u в c, е точно $\mathcal{I}(u)$. Нека u е появява точно t пъти в c. Тогава $|\mathcal{I}(u)| = 2t$. Имайки предвид, че $d(u) = |\mathcal{I}(u)|$, заключаваме, че степента на u е четно число.

Да разгледаме по-общия случай, в който u има примки. Нека u има q примки $e_1, e_2, \ldots, e_q$. Нека u се появява точно t пъти в c, както в предния случай. Очевидно, $t \geqslant q$. При наличието на примки на u има съседни появи на u в c (естествено, с ребро между тях, което ребро е някоя от примките). По-точно казано, в c има точно q подредици $u\, e_i\, u$, $1 \leqslant i \leqslant q$ (по една за всяка примка на u), като някои от тези подредици може да имат общ край u. Всяка такава триелементна подредица, отговаряща на дадена примка, има принос $+2$ към степента на u. Максималните по включване подредици са от вида $u\, e_{j_1} \cdots e_{j_r}\, u$, където $e_{j_1}, \ldots, e_{j_r} \in \{e_1, \ldots, e_q\}$, са точно $t - q$ на брой. Нека E' е множеството от всички ребра, които са вляво и вдясно от всяка от тези подредици. E' е точно множеството от ребрата, инцидентни с u, които не са примки. Тъй като тези максимални по включване подредици никога не са съседни, в смисъл, че между всеки две от тях в c има поне един връх, който не е u, очевидно $|E'| = 2(t-q)$. Имайки предвид, че $d(u)$ е сумата от $|E'|$ и $2q$ (вж. Определение 15), заключаваме, че $d(u)$ е четно число.

Теорема 21: Ойлеров път, който не е цикъл, в свързан мултиграф

G има Ойлеров път, който не е цикъл, тогава и само тогава, когато точно два върха са от нечетна степен.