NO DATE

Обхождане на графи.

1. Обхождане в ширина.

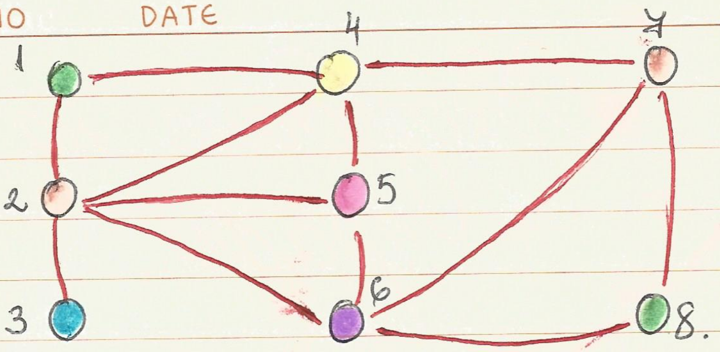
Ниво на обхождане. - подмножество на множеството от върховете на свързания граф. $G(V, E)$. В резултат на обхождането в ширина, V се разбива на нива.

Идея на обхождането.

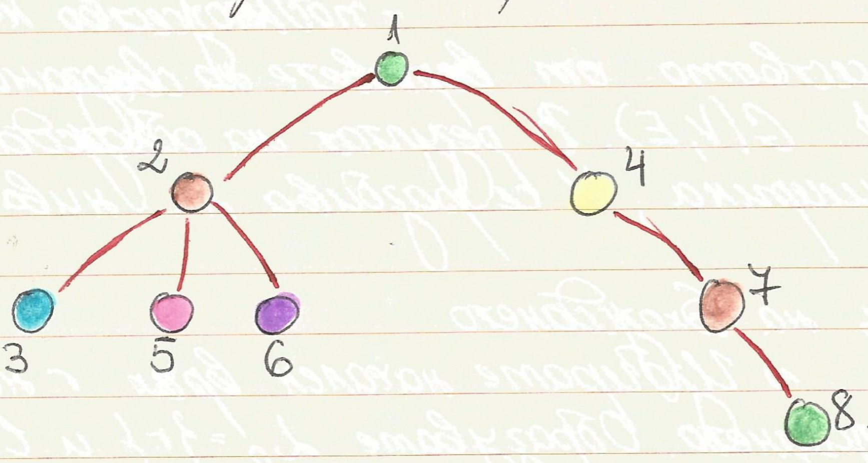
1) Избираме начален връх r на обхождането. Образуваме $L_0 = \{r\}$ и $l = 0$. и обявяваме върха r за обходен.

2) Ако $L_0 \cup L_1 \cup \dots \cup L_l = V$, тогава обхождането е завършено в противен случай преминаваме към 3.

3) Нека сме обходили върховете от нива $L_0, L_1, \dots, L_l$. Образуваме новото ниво L_{l+1} от всички необходени върхове $v_i \notin L_0 \cup L_1 \cup \dots \cup L_l$, които са съседи на върховете от L_l. Обявяваме върховете от L_{l+1} за обходени. Нека $l = l+1$ и преминаваме към 2.



В ширина спрямо 1.



NO DATE

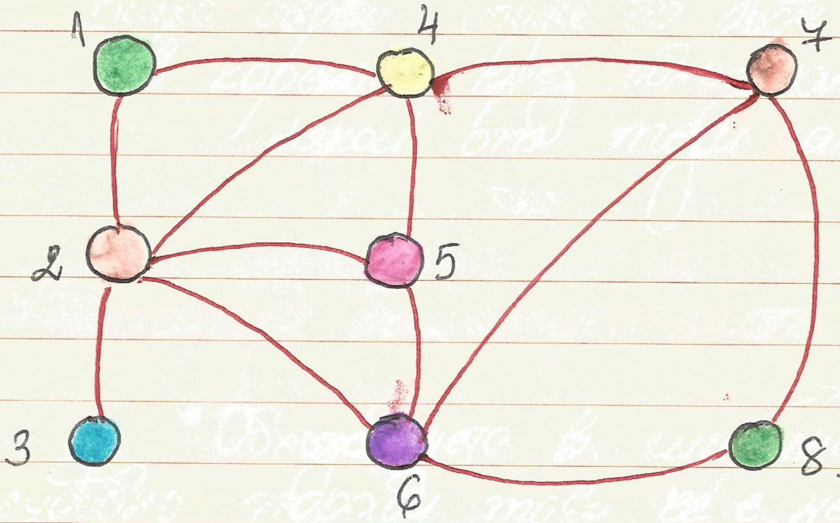
2. Обхождане в дълбочина.

t - текущ врех
$p(t)$ - бащата на върха t

Идея на обхождането:

1) „Обхождате" началния врех r.
При това $t = r$, а $p(t)$ е неопределен.

2) Търсим необходен врех, който е съседен на текущия t.
2.1) ако има такъв врех v, тогава: $p(v) = t$, $t = v$ и обявяваме v за „обходен". Преминаваме към 2.
2.2) ако няма такъв врех:
2.2.1) ако $t \neq r$, $p(t)$ е определен.
Тогава $t = p(t)$ и преминаваме към 2
(т.е. връщаме си нагоре докато не открием съседен на необходения врех)
2.2.2. ако $t = r$ обхождането е завършило.



В дълбочина спрямо 1.

- съседен на необходения (3)

NO DATE

Всяко дърво може да бъде превърнато в кореново, чрез обхождане с някой от тези алгоритми.

Недостатъци на алгоритмите:

хаби се повече памет.

* Обхождането в ширина е неудобно поради това, че е необходимо да се помнят всички обходени върхове, за да може да се построи следващото ниво.

* Обхождане в дълбочина - за него е да се пазят само върховете, разположени на пътя от началния до текущия. От друга страна, ако обхождането се прави с цел да се намери връх със зададени свойства и той се окаже на ниво с малък номер, тогава алгоритъмът в ширина ще го намери много бързо.

NO DATE

3. Ойлерови обхождания.

Ойлеров път в свързания мултиграф $G(V, E)$ наричаме ~~път~~, който ~~минава~~ съдържа еднократно ~~през~~ всяко ребро на мултиграфа и началния и крайния връх са различни.

Ойлеров цикъл - Ойлеров път, който има начало и край, които съвпадат.

Ойлеров граф - мултиграф, ребрата на който образуват Ойлеров цикъл.

ТДУ (Ойлеров цикъл):

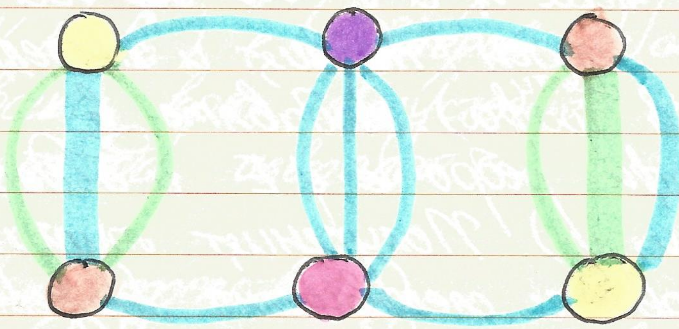
Свързаният мултиграф $G=(V, E)_{MG}$ е Ойлеров т.с.т.к. всеки връх на G е с четна степен.

Доказателство:

Доп., че графът е Ойлеров => съдържа Ойлеров цикъл. Тогава всеки връх има четна степен, защото при обхождане на мултиграфа по Ойлеров цикъл, на всяко ребро „влизащо" във връх $v_i \in V$ съответства ребро „излизащо" от v_i, а цикълът съдържа всички ребра точно по един път

Забележка:

Ако хванем произволен връх на графа, инцидентен с необходено ребро и отиваме в друг връх и правим това докато пристигнем в даден връх, а той няма инцидентни необходени ребра, това не значи, че графа е обходен.



необходени ребра

обходени ребра

NO DATE

ЧI2Y (Ойлеров път):

Свързаният мултиграф $G=(V,E,f_G)$ съдържа Ойлеров път, който не е Ойлеров цикъл, т.с.т.к. има точно 2 върха с нечетна степен.

Доказателство:

Г

1) Нека v_i и v_j са върховете с нечетна степен (първият и последният връх в мулти графа). Добавяме в мултиграфа ребро $e \notin E$ и додефинираме $f_{G'}(e) = v_i, v_j$. Полученият мултиграф G' е Ойлеров и следователно можем да построим Ойлеров цикъл. Отстраняваме добавеното ребро и получаваме път от v_i до v_j, който съдържа всички ребра на графа $G(V,E)$, точно по веднъж. и следователно е Ойлеров път.

L

2) Нека ребрата на $G=(V,E)$ образуват Ойлеров път от v_i до v_j. Добавяме реброто $e \notin E$ и додефинираме $f_{G'}(e)=(v_i, v_j)$. Пътят се превръща в Ойлеров цикъл за новополучения мултиграф G' и следователно в него всички върхове

NO DATE

са с четна степен. Добавянето на реброто (v_i, v_j) е увеличило с 1 само степените на v_i и v_j. Следователно в G всички върхове са с четна степен с изключение на тези два върха.

4.24 (Ойлеров цикъл - ориент. мултиграф):

Крайният ориентиран мултиграф $G(V, E, f_G)$ е Ойлеров, т.с.т.к за всеки връх полустепента на входа и изхода съвпадат.

Доказателство:

Нека ребрата на $G=(V, E)$ образуват Ойлеров цикъл. Тогава за всеки връх полустепента на входа и изхода съвпадат, защото графът е Ойлеров, а ако не съвпадат, то няма да могат да се обходят всички ребра.

НДУ (Ойлеров път - ориентиран мултиграф):

Крайният ориентиран мултиграф $G(V, E, f_G)$ съдържа Ойлеров път, т.с.т.к. само за два от върховете му полустепените на входа и изхода не съвпадат, като в единия връх полустепента на изхода е с единица по-голяма от полустепента на входа, а при другия обратно.

Доказателство:

Нека ребрата на ориент. мултиграф $G=(V, E, f_G)$ образуват Ойлеров път от v_i до v_j. Добавяме реброто $e \notin E$ и додефинираме $f_{G'}(e)=(v_j, v_i)$. Пътят се превръща в Ойлеров цикъл за новополучения мултиграф G' и следователно в него всички върхове имат еднаква полустепен на входа и изхода. Добавянето на реброто (v_j, v_i) е увеличило с единица само степента на изхода на v_j и полустепента на входа на v_i. Следователно в G всички върхове имат еднаква полстепен на ~~полу~~ входа и изхода, освен два от върховете,

NO DATE

за които единия има с единица по-голяма полустепен на входа отколкото на изхода, а при другия обратно.

Няма как едновременно в един граф да има Ойлеров цикъл и Ойлеров път.

Ойлеров път ≠ Ойлеров цикъл

↳ НДУ: графа да е свързан, всички върхове да имат четна степен с изключение на два от тях

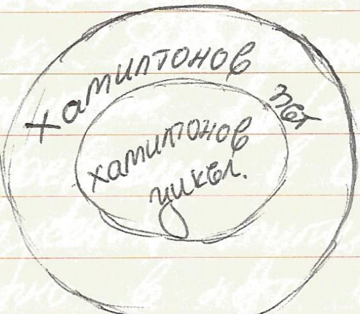
↳ НДУ: графа да е свързан и всички върхове да са с четна степен.

4. Хамилтонови обхождания.

Хамилтонов път в граф е прост път, който съдържа всички върхове, точно един път.

Хамилтонов цикъл в граф е ~~[illegible]~~ Хамилтонов път, на който края и началото съвпадат.

От Хамилтоновия цикъл => че графът има и Хамилтонов път, но обратното не е вярно.



срязващ връх - има ли такъв връх то => графа няма Хамилтонов цикъл.

Не е известен бърз алгоритъм, който да проверява дали зададен свързан граф е Хамилтонов.

Допълнителни бележки: