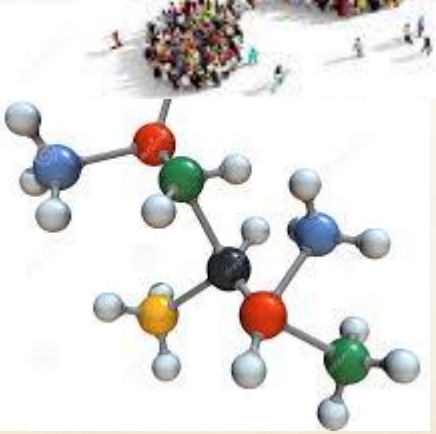
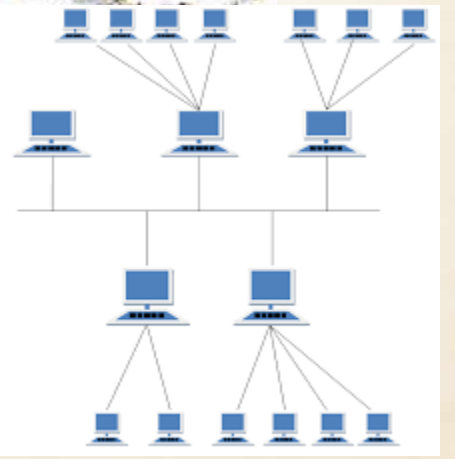
გრაფები. ცნებები და განსაზღვრებები. გრაფთა წარმოდგენა.

გრაფის განსაზღვრება

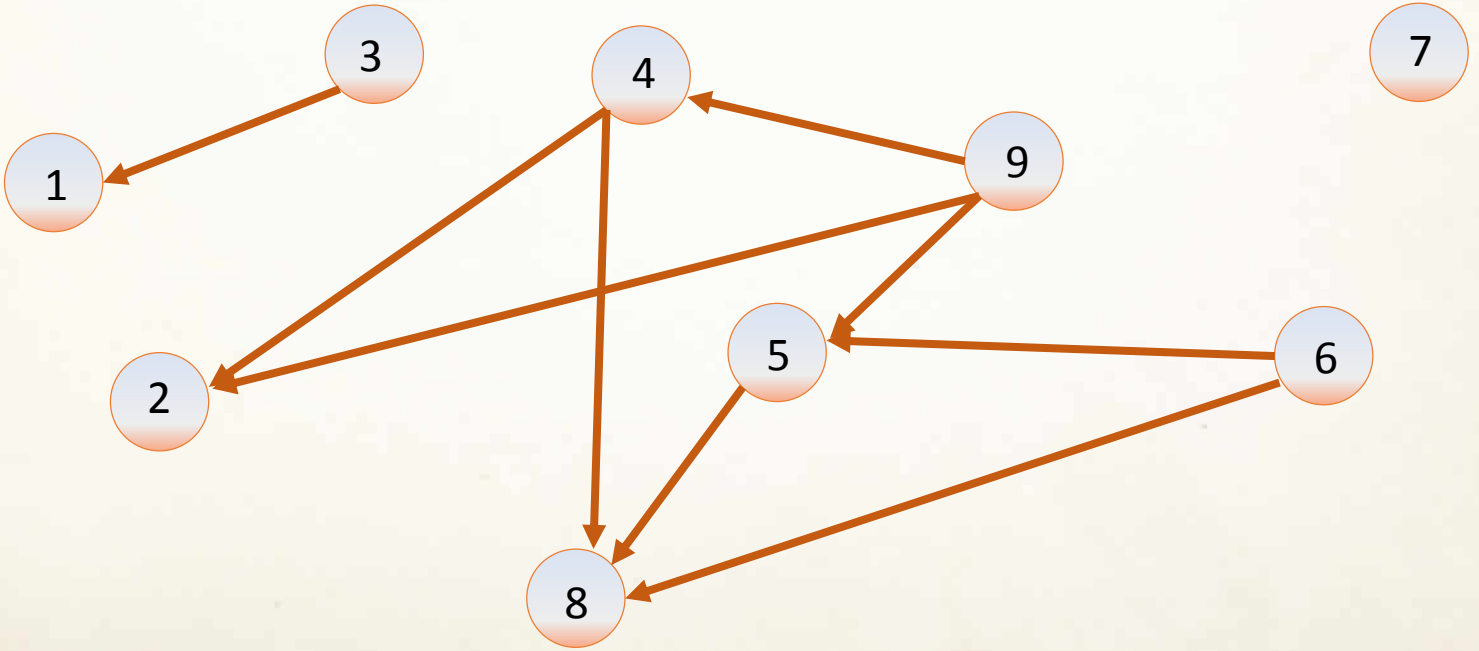
გრაფი (graph) განისაზღვრება როგორც (V,E) წყვილი, სადაც V სასრული სიმრავლეა, ხოლო E წარმოადგენს V-ს ელემენტთა ბინარულ დამოკიდებულებას, ანუ V×V სიმრავლის ქვესიმრავლეა. V სიმრავლეს უწოდებენ გრაფის **წვეროთა სიმრავლეს** (vertex set), ხოლო E-ს **წიბოთა სიმრავლეს** (edge set). მათ ელემენტებს შესაბამისად ეწოდებათ **წვერო** (vertex, მრავლობითში vertices) და **წიბო** (edge).





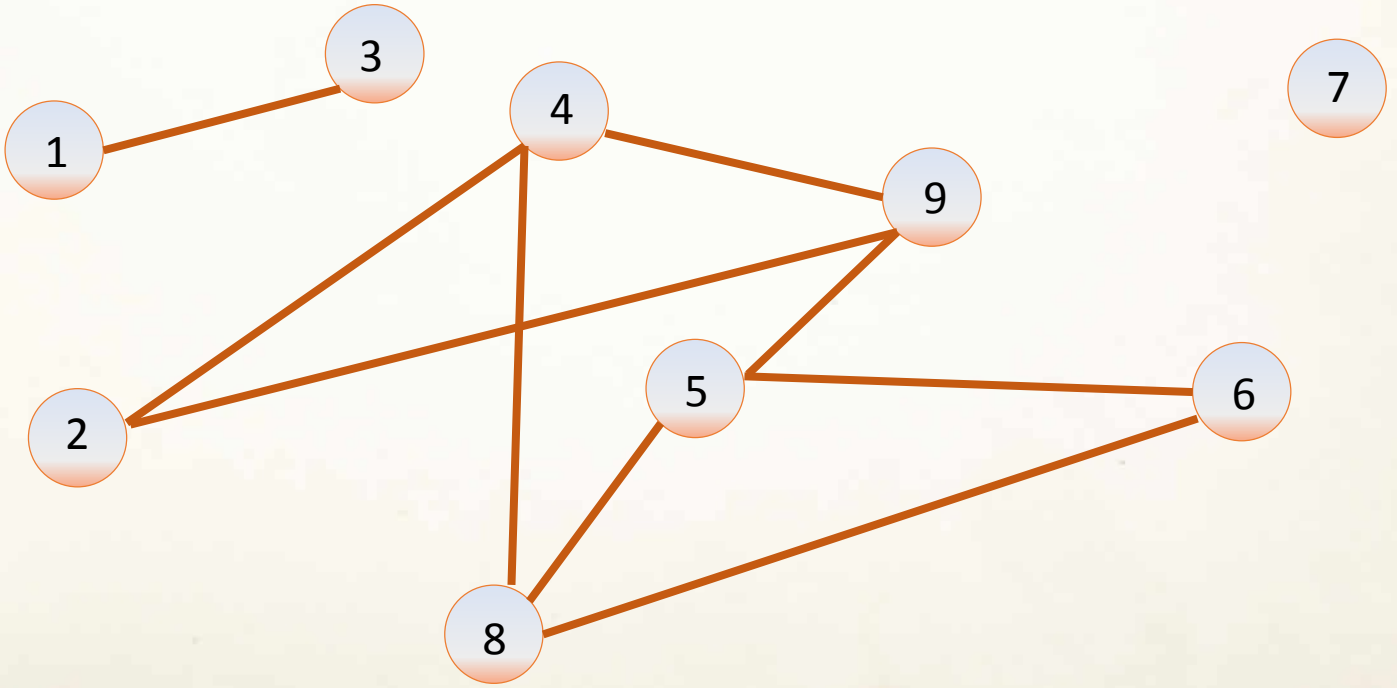
ორიენტირებული გრაფი

ორიენტირებულ (undirected) გრაფში G=(V,E) წიბოთა E სიმრავლე შედგება წვეროთა დალაგებელი (ordered) წყვილებისაგან. სხვაგვარად რომ ვთქვათ, ყველა წიბოს გააჩნია საკუთარი მიმართულება. ასეთ გრაფში (u,v) და (v,u) სხვადასხვა წიბოს აღნიშნავს, ხოლო წიბოს, რომელიც წვეროს საკუთარ თავთან აერთებს, უწოდებენ **ციკლურ წიბოს**. ორიენტირებულ გრაფს ზოგჯერ შემოკლებით **ორგრაფად** (digraph) მოიხსენიებენ.



არაორიენტირებული გრაფი

არაორიენტირებულ (undirected) გრაფში G=(V,E) წიბოთა E სიმრავლე შედგება წვეროთა დაულაგებელი (unordered) წყვილებისაგან. ასეთ გრაფში (u,v) და (v,u) ერთ და იგივე წიბოს აღნიშნავს, ხოლო ციკლურ წიბოს არ განიხილავენ.



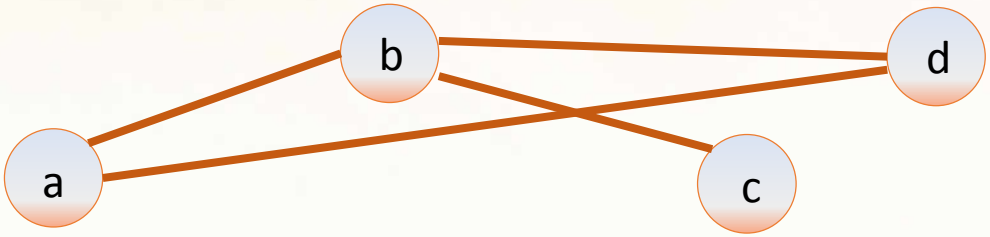
წვეროს და წიბოს ურთიერთმიმართება

(u,v) წიბოს შესახებ ორიენტირებულ გრაფში იტყვიან, რომ იგი გამოდის u წვეროდან და შედის v წვეროში.

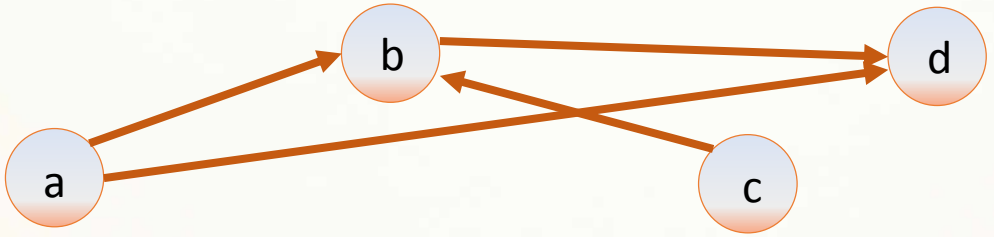
არაორიენტირებულ გრაფში (u,v) წიბოს შესახებ იტყვიან, რომ იგი u და v წვეროების **ინციდენტურია** (incident).

თუ გრაფში არსებობს (u,v) წიბო, იტყვიან, რომ v წვერო u წვეროს **მოსაზღვრეა** (adjacent). არაორიენტირებულ გრაფებში მოსაზღვრეობა სიმეტრიულია, ხოლო ორიენტირებულ გრაფებში სიმეტრიულობა აუცილებელი არ არის. თუკი ორიენტირებულ გრაფში v წვერო u წვეროს მოსაზღვრეა, წერენ u→v.

წვეროს ხარისხი



არაორიენტირებულ გრაფში წვეროს **ხარისხს** (degree) უწოდებენ ამ წვეროსადმი ინციდენტური წიბოების რაოდენობას. ნახაზზე a და d წვეროების ხარისხია 2, b წვეროს ხარისხია 3, ხოლო c წვეროს 1.



ორიენტირებულ გრაფში განასხვავებენ **შემავალ** (in-degree) და **გამომავალ** (out-degree) ხარისხებს (შესაბამისად წვეროში შემავალი და გამომავალი წიბოების რაოდენობის მიხედვით) და მათ ჯამს უწოდებენ წვეროს ხარისხს. ნახაზზე b წვეროს ხარისხია 3 (შემავალი ხარისხი – 2, გამომავალი ხარისხი – 1).

ლემა გრაფში წვეროთა ხარისხების შესახებ

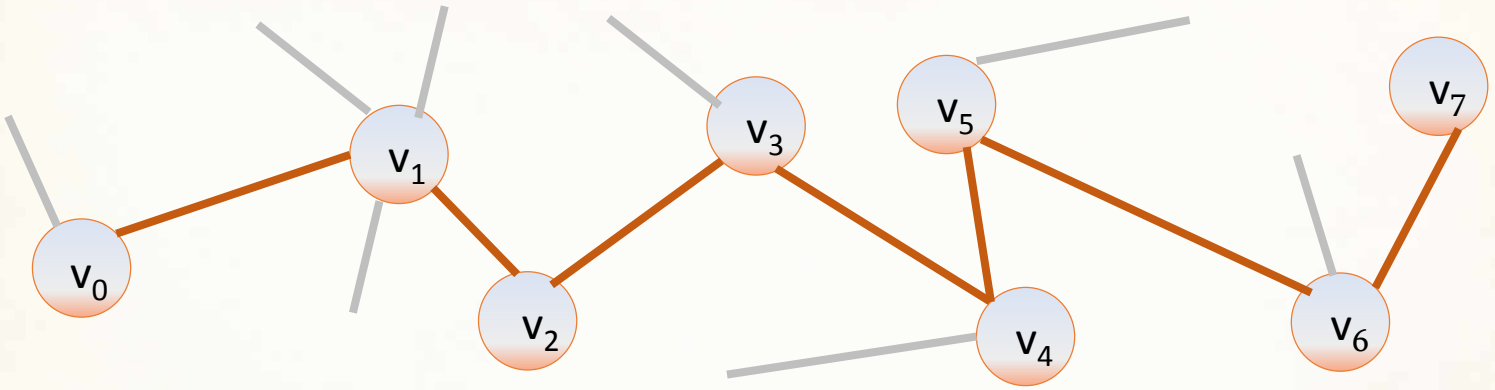
ლემა. გრაფში ყველა წვეროს ხარისხების ჯამი წიბოთა გაორმაგებული რაოდენობის ტოლია.

დამტკიცება. თუ წიბო გრაფის ორი სხვადასხვა u და v წვეროს ინციდენტურია, მაშინ წვეროთა ხარისხების ჯამის დათვლისას მას ორჯერ დავთვლით: ერთხელ u წვეროს ხარისხის დათვლისას, ხოლო მეორეჯერ v წვეროს ხარისხის დათვლისას. თუკი წიბო მარყუჟს წარმოადგენს, მასაც ორჯერ დავამატებთ ჯამში შესაბამისი წვეროს განხილვისას.

ამ ლემიდან გამომდინარეობს შედეგი:

შედეგი. ნებისმიერ გრაფში კენტი ხარისხის წვეროების რაოდენობა ლუწია

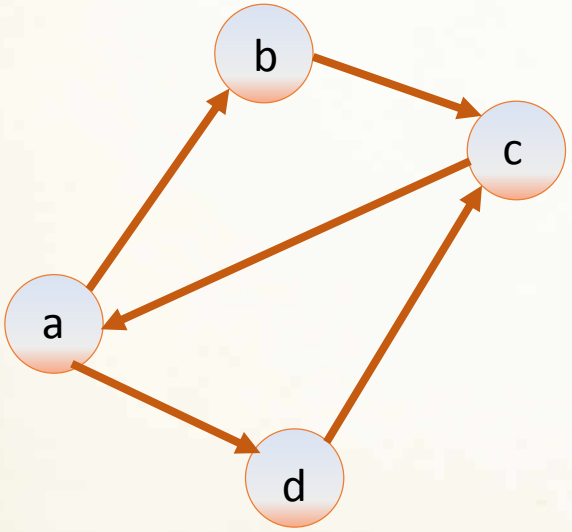
გზა გრაფში



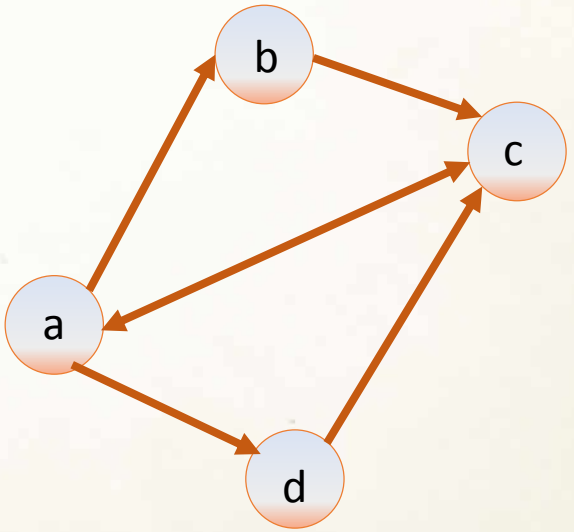
k **სიგრძის გზა** (path of lengt k) u წვეროდან v წვეროში განისაზღვრება როგორც წვეროთა $\langle v_0,v_1,v_2,\ldots,v_k\rangle$ მიმდევრობა, სადაც $v_0=u$, $v_k=v$ და $(v_{i-1},v_i)\in E$ ნებისმიერი $i=1,2,\ldots,k$-სათვის. ამრიგად k სიგრძის გზა შედგება k წიბოსაგან. იგი **შეიცავს** (contains) $v_0,v_1,v_2,..,v_k$ წვეროებს და (v_0,v_1), (v_1,v_2),...,(v_{k-1},v_k) წიბოებს. v_0 წვეროს უწოდებენ გზის **დასაწყისს**, ხოლო v_k წვეროს – გზის **ბოლოს** და ამბობენ, რომ გზა მიდის v_0-დან v_k-საკენ. თუ $v_0=v_k$, გზას უწოდებენ **ციკლურს**. თუ მოცემული u და u′ წვეროებისთვის არსებობს p გზა u-დან u′-ში, მაშინ ამბობენ, რომ u′ **მიღწევადია** u-დან p გზით (u′ is reachable from u via p). გზას ეწოდება **მარტივი** (simple), თუკი ყველა წვერო მასში განსხვავებულია.

ციკლი ორიენტირებულ გრაფში

ორიენტირებულ გრაფში **ციკლი** (cycle) ეწოდება გზას, რომელშიც საწყისი წვერო ემთხვევა ბოლო წვეროს და რომელიც ერთ წიბოს მაინც შეიცავს. ციკლს ეწოდება მარტივი თუკი მასში არ მეორდება არცერთი წვერო პირველისა და ბოლოს გარდა. გრაფს, რომელშიც არაა ციკლები, აციკლური (acyclic) ეწოდება.



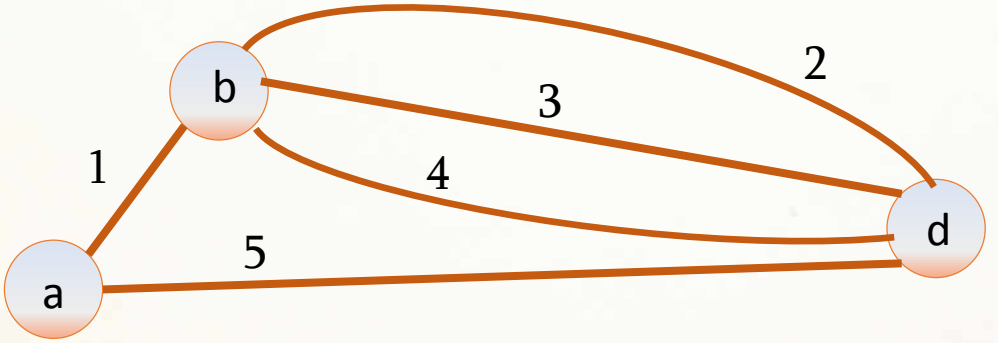
ორი ციკლი: (a,b,c,a) და (c,a,d,c)



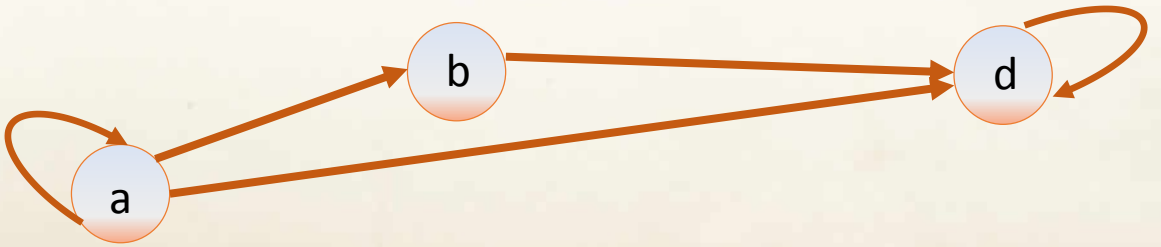
აციკლური გრაფი

ჯერადი წიბოები და მარყუჟი

თუ გრაფის ორ წვერო რამდენიმე განსხვავებული წიბოთია შეერთებული, ამ წიბოებს **ჯერად** წიბოებს უწოდებენ. გრაფში ჯერადი წიბოების არსებობისას წვეროების მიმდევრობა ცალსახად ვეღარ განსაზღვრავს გზას, ამიტომ ასეთ შემთხვევაში წიბოებს გადანომრავენ და მათ მიმდევრობას იხილავენ. ზოგ ლიტერატურაში ასეთი გზების აღსანიშნად ტერმინი „მარშრუტი“ გამოიყენება. ქვემოთ მოყვანილ ნახაზზე შესაძლოა განვიხილოთ რამდენიმე მარშრუტი a-დან b-ში: 5→4, 5→2, 5→3 ან მხოლოდ პირველი წიბო.

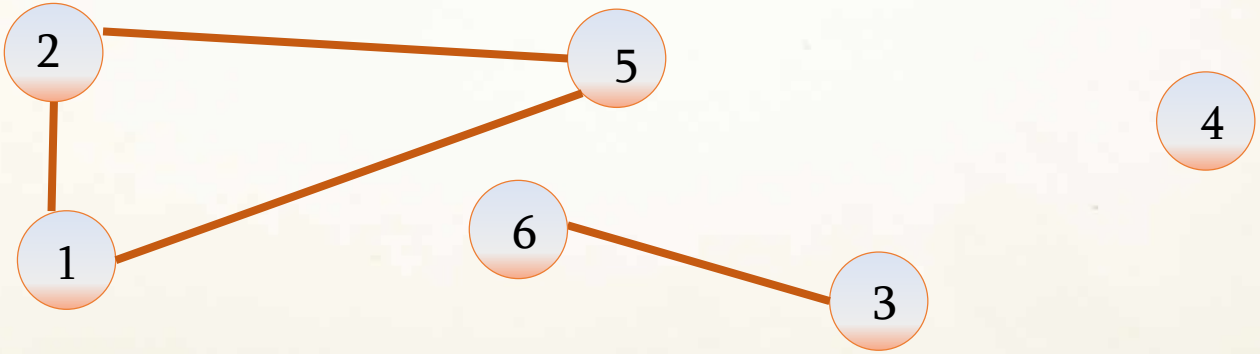


წიბოს, რომელიც წვეროს საკუთარ თავთან აერთებს **მარყუჟს** უწოდებენ.



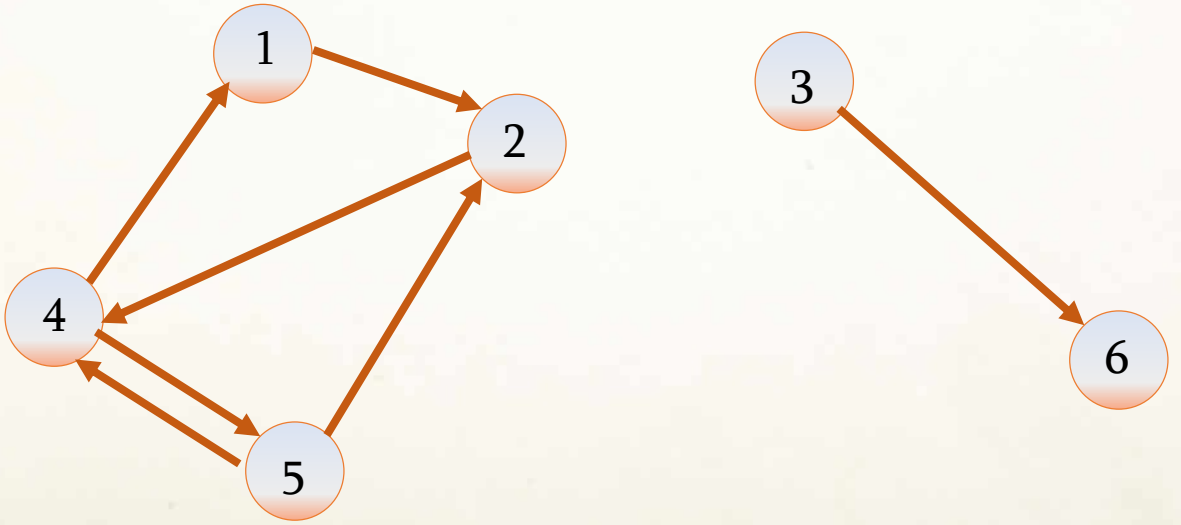
გრაფის ბმულობა

არაორიენტირებულ გრაფს ეწოდება ბმული (connected), თუკი წვეროთა ნებისმიერი წყვილისათვის არსებობს გზა ერთიდან მეორეში. არაორიენტირებულ გრაფში გზის არსებობა ერთი წვეროდან მეორეში წარმოადგენს ექვივალენტურ შესაბამისობას წვეროთა სიმრავლეზე. ექვივალენტობის კლასებს ეწოდებათ გრაფის ბმული კომპონენტები (connected components). მაგ. ნახაზზე სამი ბმული კომპონენტია: {1,2,5}, {3,6} და {4}. არაორიენტირებული გრაფი ბმულია მაშინ და მხოლოდ მაშინ, როცა ის შედგება ერთადერთი ბმული კომპონენტისაგან.



გრაფის ბმულობა

ორიენტირებულ გრაფს ეწოდება **ძლიერად ბმული** (strongly connected), თუკი მისი ნებისმიერი წვეროდან მიღწევადია (ორიენტირებული გზებით) ნებისმიერი სხვა წვერო. ნებისმიერი ორიენტირებული გრაფი შეიძლება დაიყოს **ძლიერად ბმულ კომპონენტებად** (strongly connected components). ნახაზზე არის სამი ასეთი კომპონენტი {1,2,4,5}, {3} და {6}. შევნიშნოთ, რომ 3 და 6 წვეროები ერთად არ ჰქმნიან ძლიერად ბმულ კომპონენტს, რადგან არ არსებობს გზა 6-დან 3-ში.



მანძილი გრაფის წვეროებს შორის

მარშრუტის სიგრძე ეწოდება მასში შემავალი წიბოების რაოდენობას. **მანძილი** გრაფის u და v წვეროებს შორის ეწოდება (u, v)-მარშრუტებს შორის უმცირესს და აღინიშნება, როგორც ρ(u,v). თუ u და v წვეროები ბმული არ არის, მაშინ მათ შორის მანძილი უსასრულობად ითვლება. რადგან ტრივიალური მარშრუტის სიგრძე ითვლება 0-ად, ρ(u, u)=0.
გრაფის ნებისმიერი წვეროებისთვის სრულდება პირობები:

$$\rho(u, v) = \rho(v, u) \text{ და}$$

$$\rho(u, v) + \rho(v, w) > \rho(u, w) \text{ (სამკუთხედის უტოლობა)}$$

წონადი გრაფის შემთხვევაში მანძილი გრაფის ორ წვეროს შორის განისაზღვრება, როგორც შესაბამისი წიბოების წონათა ჯამი.

გრაფის დიამეტრი, რადიუსი და ცენტრი

ვთქვათ G ბმული გრაფია. G გრაფის **დიამეტრი** ეწოდება ამ გრაფის წვეროთა ყველა წყვილს შორის მანძილის მაქსიმუმს და აღინიშნება d(G)-თი. თუ v გრაფის წვეროა, r(v)-თი აღინიშნება მაქსიმალური მანძილი v წვეროდან დანარჩენ წვეროებამდე არსებულ მანძილებს შორის. გრაფის **რადიუსს** უწოდებენ მინიმალურს r(v) რიცხვებს შორის, სადაც v ღებულობს მნიშვნელობებს V(G) სიმრავლიდან. G გრაფის რადიუსი აღინიშნება r(G)-თი. ნებისმიერი w წვეროს, რომლისთვისაც r(w) = r(G), ეწოდება გრაფის **ცენტრი**.

გამოყენების მაგალითი

ვთქვათ, გრაფი წარმოადგენს გზების ქსელს, სადაც წვეროებს წარმოადგენს დასახლებული პუნქტები, ხოლო წიბოებს - მათ შორის არსებული გზები. საჭიროა სკოლების, საავადმყოფოების და სხვა მომსახურების ობიექტების ოპტიმალურად განლაგება. ოპტიმალურობის ერთ-ერთ კრიტერიუმად ითვლება მანძილების მინიმიზაცია ამ ობიექტებიდან ყველაზე დაშორებულ პუნქტებამდე. მაგრამ მანძილი v წვეროდან ყველაზე დაშორებულ წერტილამდე არის r(v), ანუ ობიექტები უნდა განლაგდნენ ისეთ v წვეროებში, რომელთა r(v) მინიმალურია, ანუ გრაფის ცენტრებში.

თეორემა გრაფის რადიუსისა და დიამეტრის შესახებ

თეორემა. თუ **G** ბმული გრაფია, მაშინ სრულდება უტოლობა:

$$d(G)/2 \leq r(G) \leq d(G).$$

დამტკიცება. უტოლობა **r(G) ≤ d(G)** გამომდინარეობს განსაზღვრებიდან:

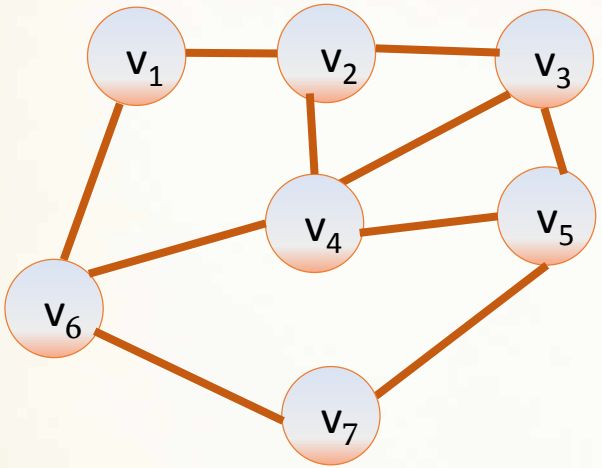
$$r(G) = \min_{v \in V(G)} r(v) \leqslant \max_{v \in V(G)} r(v) = \max_{v \in V(G)} \max_{u \in V(G)} \rho(u, v) = d(G).$$

d(G)/2 ≤ r(G) უტოლობის დასამტკიცებლად საკმარისია ვაჩვენოთ, რომ გრაფის ნებისმიერ u და v წვეროებს შორის მანძილი არ აღემატება 2r-ს. ვთქვათ w გრაფის ნებისმიერი ცენტრია, მაშინ ცენტრის განსაზღვრებისა და სამკუთხედის უტოლობიდან მივიღებთ:

$$\rho(u, v) \leqslant \rho(u, w) + \rho(w, v) \leqslant r + r = 2r,$$

რისი დამტკიცებაც გვინდოდა.

მაგალითი



	v_1	v_2	v_3	v_4	v_5	v_6	v_7	$r(v_i)$
v_1	0	1	2	2	3	1	2	3
v_2	1	0	1	1	2	2	3	3
v_3	2	1	0	1	1	2	2	2
v_4	2	1	1	0	1	1	2	2
v_5	3	2	1	1	0	2	1	3
v_6	1	2	2	1	2	0	1	2
v_7	2	3	2	2	1	1	0	3

შევადგინოთ ცხრილი, სადაც შესაბამისი წვეროების გადაკვეთაზე ჩავწერთ მანძილს ამ წვეროებს შორის. უკანასკნელ სვეტში ჩავწეროთ $r(v_i)$-ის მნიშვნელობები. ცხადია, რომ გრაფის დიამეტრი იქნება მაქსიმალური რიცხვი ცხრილის ბოლო სვეტში, ხოლო რადიუსი - მინიმალური. ამრიგად, ნახაზზე მოცემული გრაფის დიამეტრია 3, ხოლო რადიუსი - 2, ხოლო გრაფის ცენტრებს წარმოადგენენ წვეროები v_3, v_4 და v_6.

გრაფთა წარმოდგენა

პროგრამირებაში არსებობს G=(V,E) გრაფის წარმოდგენის ორი სტანდარტული მეთოდი:

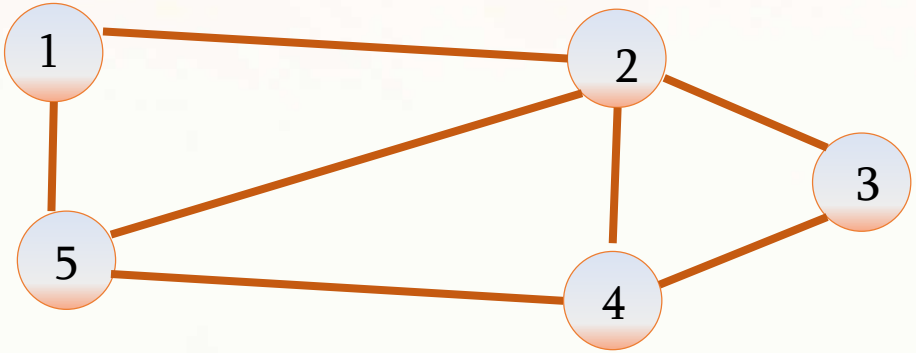
ა) **მოსაზღვრე წვეროების სიათა** (adjancency-list representation) ჩამონათვალით.

ბ) **მოსაზღვრეობის მატრიცით** (adjancency matrix).

პირველი მეთოდი უფრო მოსახერხებელია **ხალვათი** (sparse) გრაფებისათვის, სადაც |E| მკვეთრად მცირეა |V|2-თან შედარებით, ხოლო მეორე მეთოდი უფრო ეფექტურია **მკვრივი** (dense) გრაფებისათვის, სადაც |E| უახლოვდება |V|2-ს.

G=(V,E) გრაფის წარმოდგენა მოსაზღვრე წვეროების სიებით იყენებს |V| ცალი სიისაგან შემდგარ Adj მასივს. თითოეული u∈V წვეროსათვის Adj(u) შეიცავს u-ს მოსაზღვრე წვეროთა სიას ნებისმიერი მიმდევრობით. C++-ში რეალიზაცია შესაძლებელია vector-ის საშუალებით;

გრაფთა წარმოდგენა

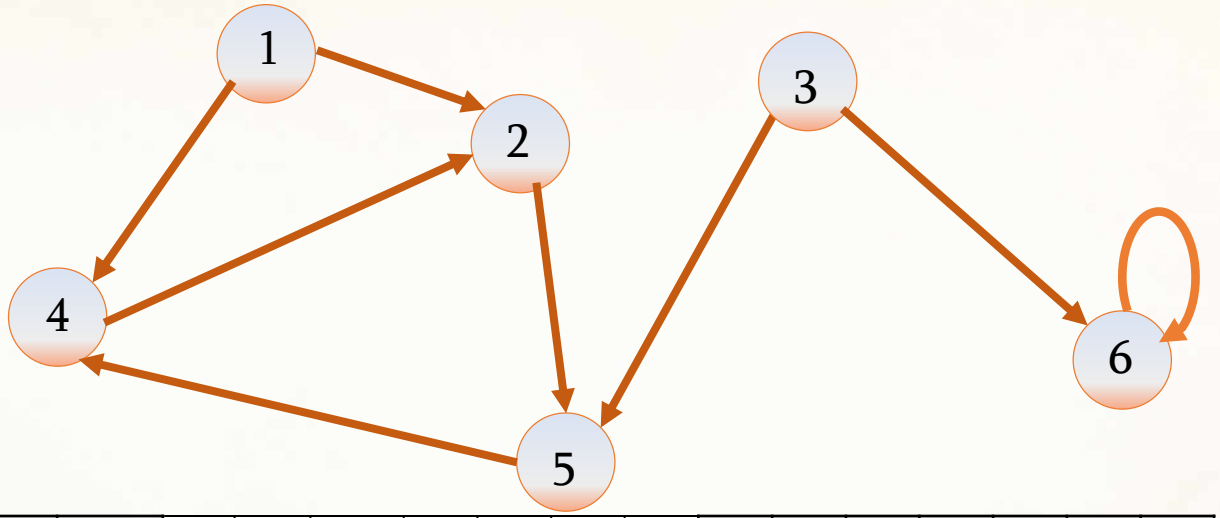


1	2	5			
2	1	5	3	4	
3	2	4			
4	2	5	3		
5	4	1	2		

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

ნახაზზე მოცემულია გრაფთა წარმოდგენის ორივე მეთოდი არაორიენტირებული გრაფებისათვის.

გრაფთა წარმოდგენა



1	2	4						
2	5							
3	6	5						
4	2							
5	4							
6	6							

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

ნახაზზე მოცემულია გრაფთა წარმოდგენის ორივე მეთოდი ორიენტირებული გრაფებისათვის.

გრაფთა წარმოდგენა

მოსაზღვრე წვეროთა ყველა სიის სიგრძეთა ჯამი ორიენტირებული გრაფისათვის წიბოთა რაოდენობის ტოლია, ხოლო არაორიენტირებული გრაფისათვის — წიბოთა გაორმაგებული რაოდენობის ტოლი, რადგან (u,v) წიბო წარმოშობს ელემენტს როგორც u, ასევე v წვეროს მოსაზღვრე წვეროთა სიაში. ორივე შემთხვევაში მეხსიერების საჭირო მოცულობაა O(V+E). ამ მეთოდით შესაძლებელია **წონადი** გრაფების (weighted graphs) შენახვაც, სადაც ყოველ წიბოს მიწერილი აქვს რაღაც ნამდვილი **წონა** (weight), ანუ მოცემულია **წონითი ფუნქცია** (weight function) w: E→R. ამ შემთხვევაში მოსახერხებელია (u,v)∈E წიბოს w(u,v) წონის შენახვა v წვეროსთან ერთად u წვეროს მოსაზღვრე წვეროთა სიაში. თუმცა მეთოდს აქვს მნიშვნელოვანი ნაკლი: u-დან v-ში წიბოს არსებობის დასადგენად, საჭიროა გადავამოწმოთ მთლიანი სია Adj(u) მასში v-ს მოსაძებნად. ძებნას შესაძლოა თავი ავარიდოთ, თუ გამოვიყენებთ მოსაზღვრეობის მატრიცას, თუმცა ამ შემთხვევაში მეტი მანქანური მეხსიერებაა საჭირო.

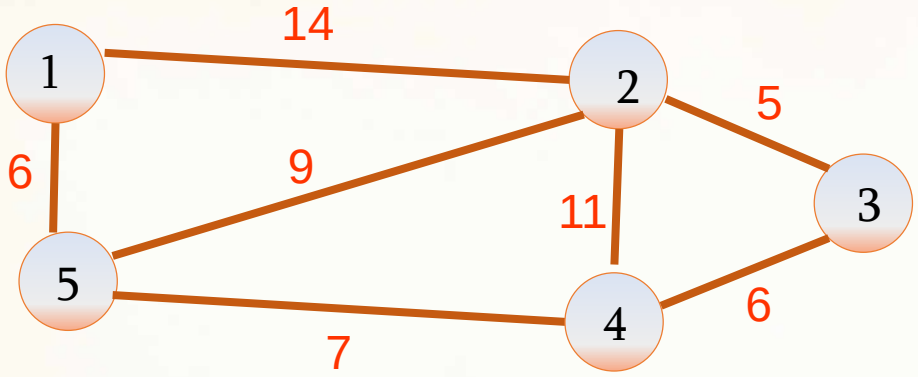
გრაფთა წარმოდგენა

მოსაზღვრეობის მატრიცა არაორიენტირებული გრაფებისათვის სიმეტრიულია (ანუ $a_{ij}=a_{ji}$ ნებისმიერი I,j=1,2,3,…,n-სათვის).

მოსაზღვრეობის მატრიცის მთავარი უპირატესობა იმაში მდგომარეობს, რომ ერთი ოპერაციით შეგვიძლია დავადგინოთ, არსებობს თუ არა ინციდენტური წიბო მოცემული ორი წვეროსათვის. გარდა ამისა, თუ გრაფს გააჩნია რაიმე ალგებრული თვისებები, მაშინ შესაძლებელია მატრიცული ალგებრის გამოყენება (მაგალითად, მატრიცის ახარისხება)

მოსაზღვრეობის მატრიცის მთავარი ნაკლი გამოყენებული მეხსიერების დიდი მოცულობაა. გარდა ამისა, გრაფებზე მომუშავე ალგორითმების დიდი ნაწილი მოითხოვს მეზობელი წვეროების ჩამონათვალს, რაც ხალვათ გრაფებში გამოიწვევს ბევრ უსარგებლო შემოწმებას.

გრაფთა წარმოდგენა



1	2	14
5	4	7
5	2	9
1	5	6
2	4	11
4	3	6
2	3	5

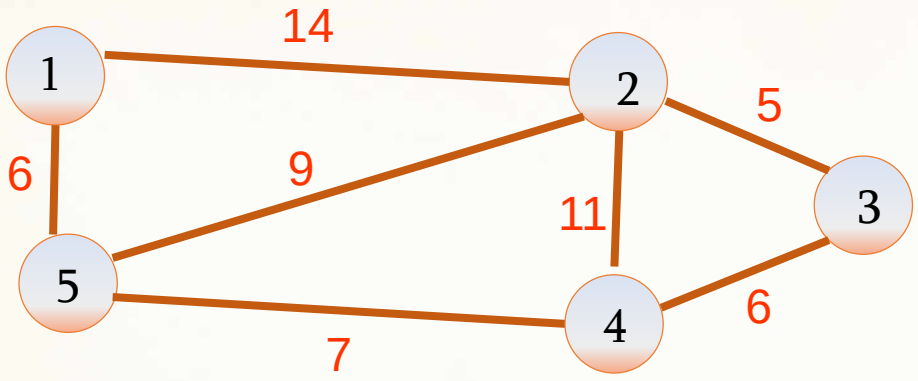
დამხმარე მასივი

1	2	3	4	5
1	2	0	0	0

ბოლო ინდექსი=6

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	0	0	0	0	0	0	0	0	0	0	0	0
14	14	0	0	0	0	0	0	0	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

გრაფთა წარმოდგენა



1 2 14
5 4 7
5 2 9
1 5 6
2 4 11
4 3 6
2 3 5

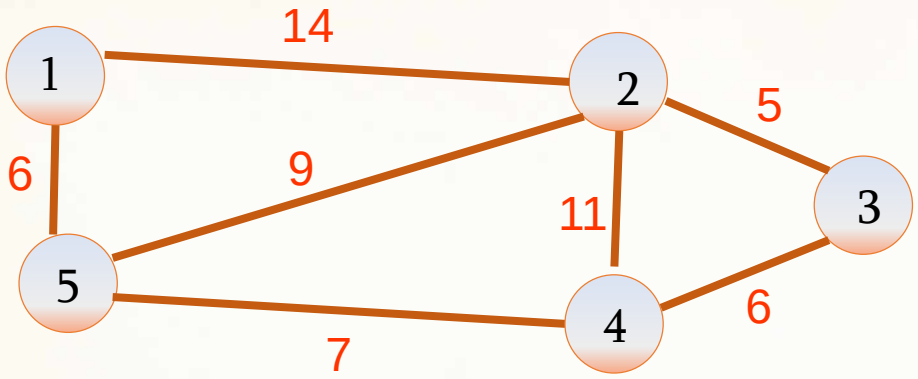
დამხმარე მასივი

1	2	3	4	5
1	2	0	4	5

ბოლო ინდექსი=6

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	0	5	4	0	0	0	0	0	0	0	0	0
14	14	0	7	7	0	0	0	0	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

გრაფთა წარმოდგენა



1 2 14
5 4 7
5 2 9
1 5 6
2 4 11
4 3 6
2 3 5

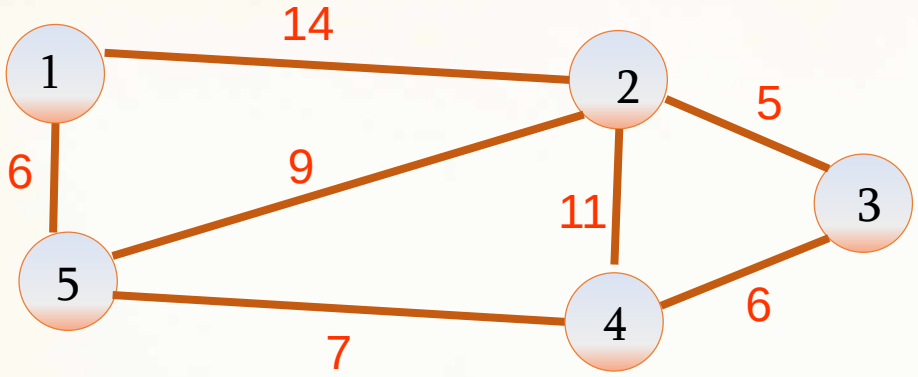
დამხმარე მასივი

1	2	3	4	5
1	7	0	4	6

ბოლო ინდექსი=8

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	0	5	4	2	5	0	0	0	0	0	0	0
14	14	0	7	7	9	9	0	0	0	0	0	0	0
-1	7	-1	-1	6	-1	-1	-1	-1	-1	-1	-1	-1	-1

გრაფთა წარმოდგენა



1 2 14
5 4 7
5 2 9
1 5 6
2 4 11
4 3 6
2 3 5

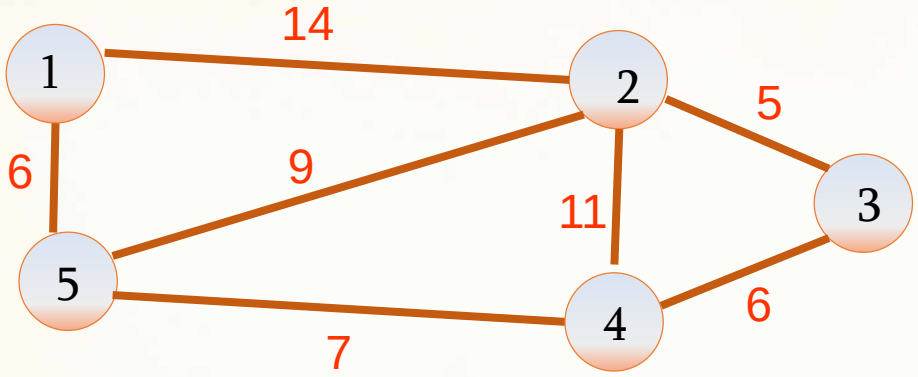
დამხმარე მასივი

1	2	3	4	5
8	7	0	4	9

ბოლო ინდექსი=10

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	0	5	4	2	5	5	1	0	0	0	0	0
14	14	0	7	7	9	9	6	6	0	0	0	0	0
8	7	-1	-1	6	9	-1	-1	-1	-1	-1	-1	-1	-1

გრაფთა წარმოდგენა



1 2 14
5 4 7
5 2 9
1 5 6
2 4 11
4 3 6
2 3 5

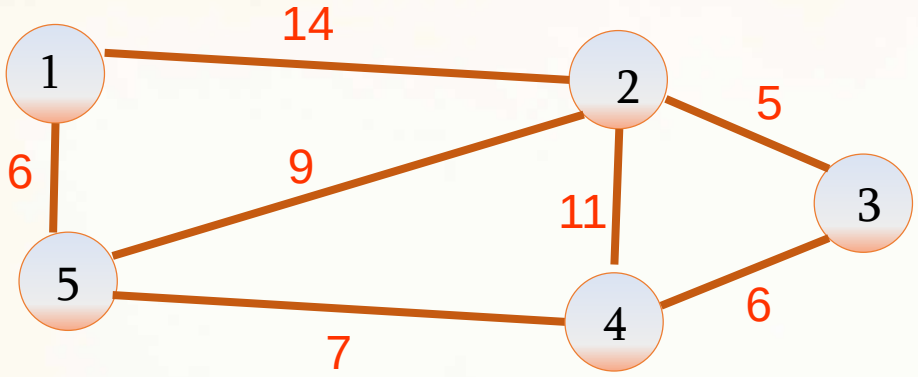
დამხმარე მასივი

1	2	3	4	5
8	10	0	11	6

ბოლო ინდექსი=12

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	0	5	4	2	5	5	1	4	2	0	0	0
14	14	0	7	7	9	9	6	6	11	11	0	0	0
8	7	-1	11	6	9	10	-1	-1	-1	-1	-1	-1	-1

გრაფთა წარმოდგენა



1 2 14
5 4 7
5 2 9
1 5 6
2 4 11
4 3 6
2 3 5

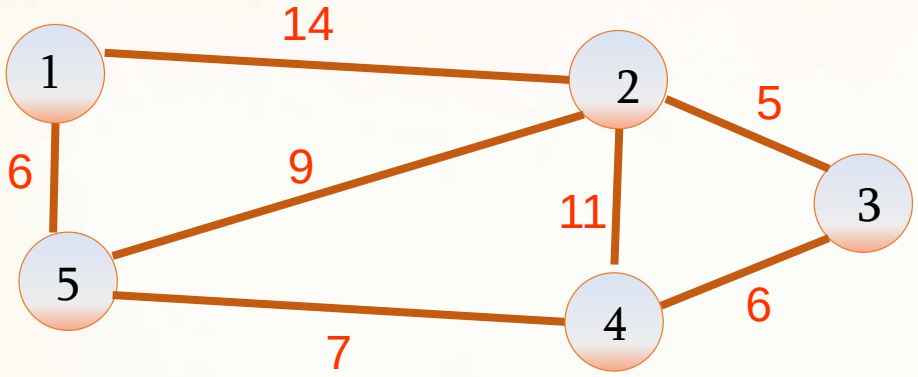
დამხმარე მასივი

1	2	3	4	5
8	10	3	12	6

ბოლო ინდექსი=13

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	4	5	4	2	5	5	1	4	2	3	0	0
14	14	6	7	7	9	9	6	6	11	11	6	0	0
8	7	-1	11	6	9	10	-1	-1	-1	12	-1	-1	-1

გრაფთა წარმოდგენა



1 2 14
5 4 7
5 2 9
1 5 6
2 4 11
4 3 6
2 3 5

დამხმარე მასივი

1	2	3	4	5
8	10	3	12	6

ბოლო ინდექსი=15

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	4	5	4	2	5	5	1	4	2	3	3	2
14	14	6	7	7	9	9	6	6	11	11	6	5	5
8	7	14	11	6	9	10	-1	-1	13	12	-1	-1	-1

ამოცანები

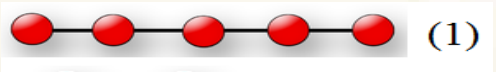
ქსელის ტოპოლოგია

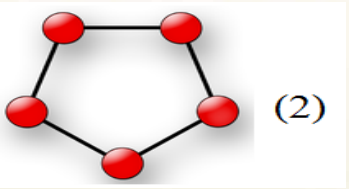
codeforces.com/problemset/problem/292/B

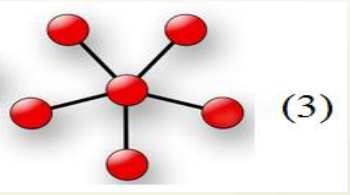
იმ ორგანიზაციის ქსელი, სადაც პოლიკარპი სისტემურ ადმინისტრატორად მუშაობს, შედგება *n* კომპიუტერისაგან, რომლებიც გადანომრილია 1-დან *n*-მდე. ცნობილია, რომ კომპიუტერების ნებისმიერი წყვილი ერთმანეთს უკავშირდება ან უშუალოდ ან სხვა კომპიუტერების გავლით.

პოლიკარპს სურს გაარკვიოს ქსელის ტოპოლოგია. ქსელის ტოპოლოგიას უწოდებენ კომპიუტერთა შეერთების სქემას და პოლიკარპისათვის ცნობილია სამი ასეთი სქემა: წრფე, რგოლი და ვარსკვლავი. წრფივი ტოპოლოგიის დროს არსებობს ერთი მავთული, რომელზეც მიერთებულია ყველა კომპიუტერი. რგოლური ტოპოლოგიისას თითოეული კომპიუტერი უერთდება ორ სხვას. ვარსკვლავური ტოპოლოგიის დროს ყველა კომპიუტერი უერთდება ერთ მთავარ კომპიუტერს.

თუ ქსელს წარმოვიდგენთ არაორიენტირებული ბმული გრაფის სახით, მაშინ „წრფე“ იქნება ერთადერთი გზისაგან შედგენილი ბმული გრაფი, რომელშიც კიდურა წვეროების გარდა, ყველა წვერო შეერთებულიაორ სხვა წვეროსთან. „რგოლში“ ყველა წვერო შეერთებულია ორ სხვა წვეროსთან, ხოლო „ვარსკვლავში“ ერთი წვერო უერთდება ყველა დანარჩენს. ამ ტოპოლოგიებში უკეთ გასარკვევად იხილეთ ნახაზი.







თქვენ გეძლევათ ბმული არაორიენტირებული გრაფი, რომელიც აღწერს კომპიუტერულ ქსელს და დაწერეთ პროგრამა, რომელიც გაარკვევს ამ ქსელის ტოპოლოგიას.

შესატანი მონაცემები: პირველ სტრიონში მოცემულია ორი მთელი რიცხვი n და m ($4 \le n \le 10^5$; $3 \le m \le 10^5$) — შესაბამისად წვეროების და წიბოების რაოდენობა გრაფში. მომდევნო m სტრიქონიდან თითოეულში მოცემულია წიბოების აღწერა. i-ურ სტრიქონში მოცემულია მთელი რიცხვების წყვილი x_i, y_i ($1 \le x_i, y_i \le n$), რომელიც აღწერს i-ურ წიბოს. გარანტირებულია, რომ გრაფი ბმულია და მასში არ არსებობენ ჯერადი წიბოები და მარყუჟები.

გამოსატანი მონაცემები: ერთადერთ სტრიქონში გამოიტანეთ ქსელის ტოპოლოგიის დასახელება. თუ ტოპოლოგია წრფივია, გამოიტანეთ «bus topology», რგოლის შემთხვევაში გამოიტანეთ «ring topology» და ვარსკვლავის შემთხვევაში გამოიტანეთ «star topology». თუ ქსელის აგებულება არცერთ მათგანს არ ემთხვევა, გამოიტანეთ «unknown topology» . თითოეული პასუხი გამოიტანეთ ბრჭყალების გარეშე.

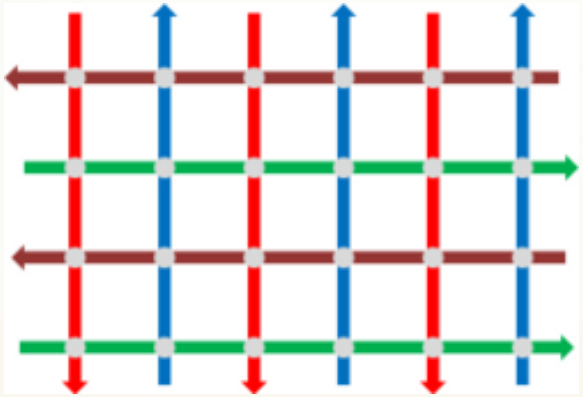
შესატანი მონაცემები			
4 3 1 2 2 3 3 4	4 4 1 2 2 3 3 4 4 1	4 3 1 2 1 3 1 4	4 4 1 2 2 3 3 1 1 4
გამოსატანი მონაცემები			
bus topology	ring topology	star topology	unknown topology

ამოცანები

ძლიერად ბმული ქალაქი

http://codeforces.com/problemset/problem/475/B

განვიხილოთ ქალაქი, რომელშიც *n* ჰორიზონტალური ქუჩა იკვეთება *m* ვერტიკალურ ქუჩასთან და ფორმირდება ბადე ზომით (*n* - 1) × (*m* - 1). საცობების თავიდან აცილების მიზნით, ქალაქის მერიამ გადაწყვიტა, რომ თითოეულ ქუჩაზე მოძრაობა გახდეს ცალმხრივი. ე. ი. ჰორიზონტალურ ქუჩებზე მოძრაობა შეიძლება ან მხოლოდ დასავლეთიდან აღმოსავლეთით, ან მხოლოდ აღმოსავლეთიდან დასავლეთით, ხოლო ვერტიკალურ ქუჩებზე მოძრაობა შეიძლება ან მხოლოდ ჩრდილოეთიდან სამხრეთით, ან მხოლოდ სამხრეთიდან ჩრდილოეთით. ნებისმიერ გზაჯვარედინზე შეიძლება გადასვლა ჰორიზონტალური ქუჩიდან ვერტიკალურზე და პირიქით.



ქალაქის მერმა მიიღო ქუჩებში მოძრაობის მიმართულების რამდენიმ ვარიანტი. დაწერეთ პროგრამა, რომელიც გაარკვევს, შეიძლება თუ არა, მოცემულ ვარიანტში მივაღწიოთ ნებისმიერი გზაჯვარედინიდან ნებისმიერ სხვა გზაჯვარედინამდე, თუ ვიმოძრავებთ ვარიანტში მითითებული მიმართულებებით.

შესატანი მონაცემები: პირველ სტრიქონში მოცემულია ორი მთელი რიცხვი *n*, *m*, (2 ≤ *n*, *m* ≤ 20), რომლებიც შესაბამისად აღნიშნავენ ჰორიზონტალური და ვერტიკალური გზების რაოდენობას.
მეორე სტრიქონში მოცემულია *n* ცალი სიმბოლო '<' და '>' — ჰორიზონტალური ქუჩების მიმართულება. '<' სიმბოლოს შესაბამისი ქუჩა მიმართულია აღმოსავლეთისაგან დასავლეთისაკენ, '>' სიმბოლოს შემთხვევაში - პირიქით. ქუჩები ჩამოთვლილია ჩრდილოეთიდან სამხრეთისაკენ.
მესამე სტრიქონში მოცემულია *m* ცალი სიმბოლო '^' და 'v' — ვერტიკალური ქუჩების მიმართულება. '^' სიმბოლოს შესაბამისი ქუჩა მიმართულია სამხრეთიდან ჩრდილოეთისკენ, 'v' სიმბოლოს შემთხვევაში - პირიქით. ქუჩები ჩამოთვლილია დასავლეთიდან აღმოსავლეთისაკენ.
გამოსატანი მონაცემები: თუ მოცემული სქემა პასუხობს მოთხოვნებს, გამოიტანეთ "YES", წინააღმდეგ შემთხვევაში - "NO".
მაგალითები
შესატანი მონაცემები:
3 3
><>
v^v
გამოსატანი მონაცემები
NO
შესატანი მონაცემები:
4 6
<><>
v^v^v^
გამოსატანი მონაცემები
YES