

Граф

- $G = (V, E)$
- V – оройн олонлог.
- Оройг бас зангилаа, цэг(пункт) гэж нэрлэдэг.
- E – ирмэг(нуруу)-ийн олонлог.
- Ирмэг хоёр өөр оройг холбодог.
- Ирмэгийг бас нум, шулуун гэж нэрлэдэг.
- Чиглэлтэй ирмэг зүг чигийг заадаг (u, v) .



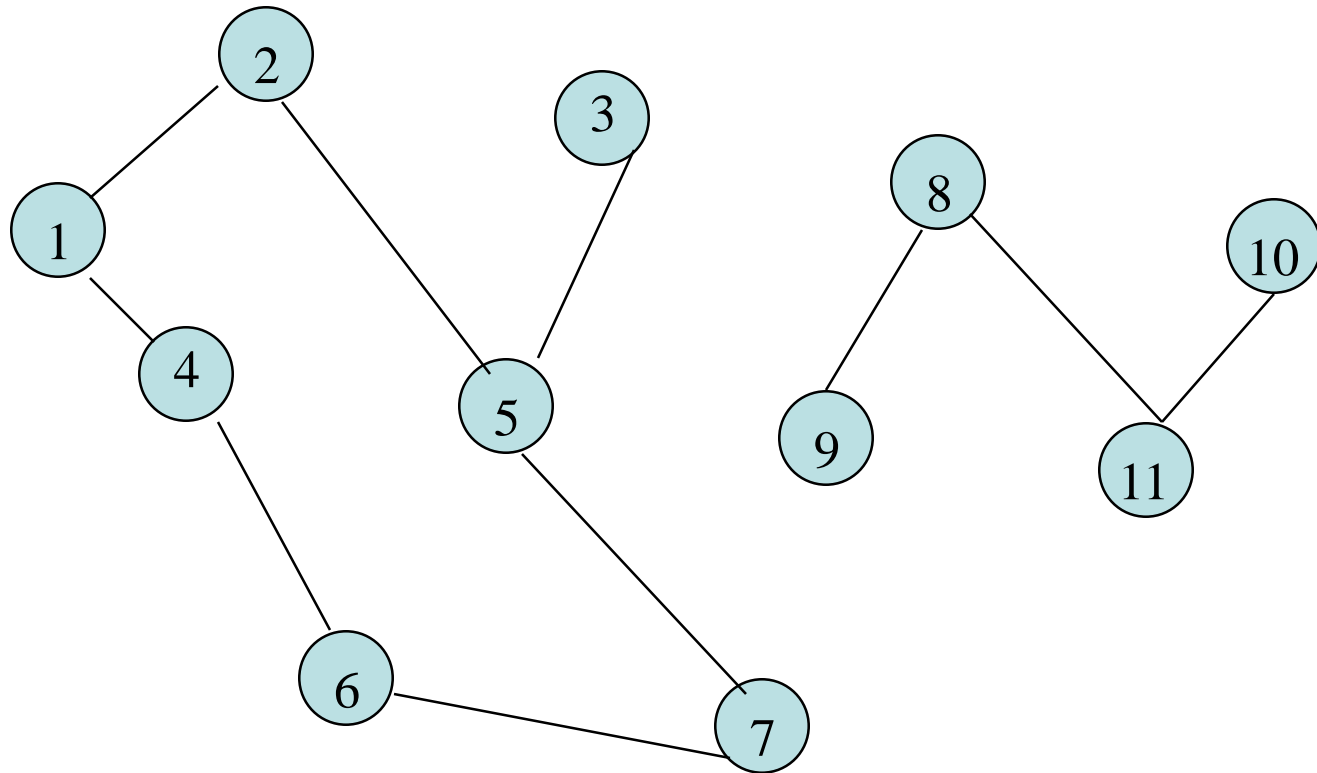
Граф

- Чиглэлгүй ирмэг зүг чиг заахгүй (u,v) .

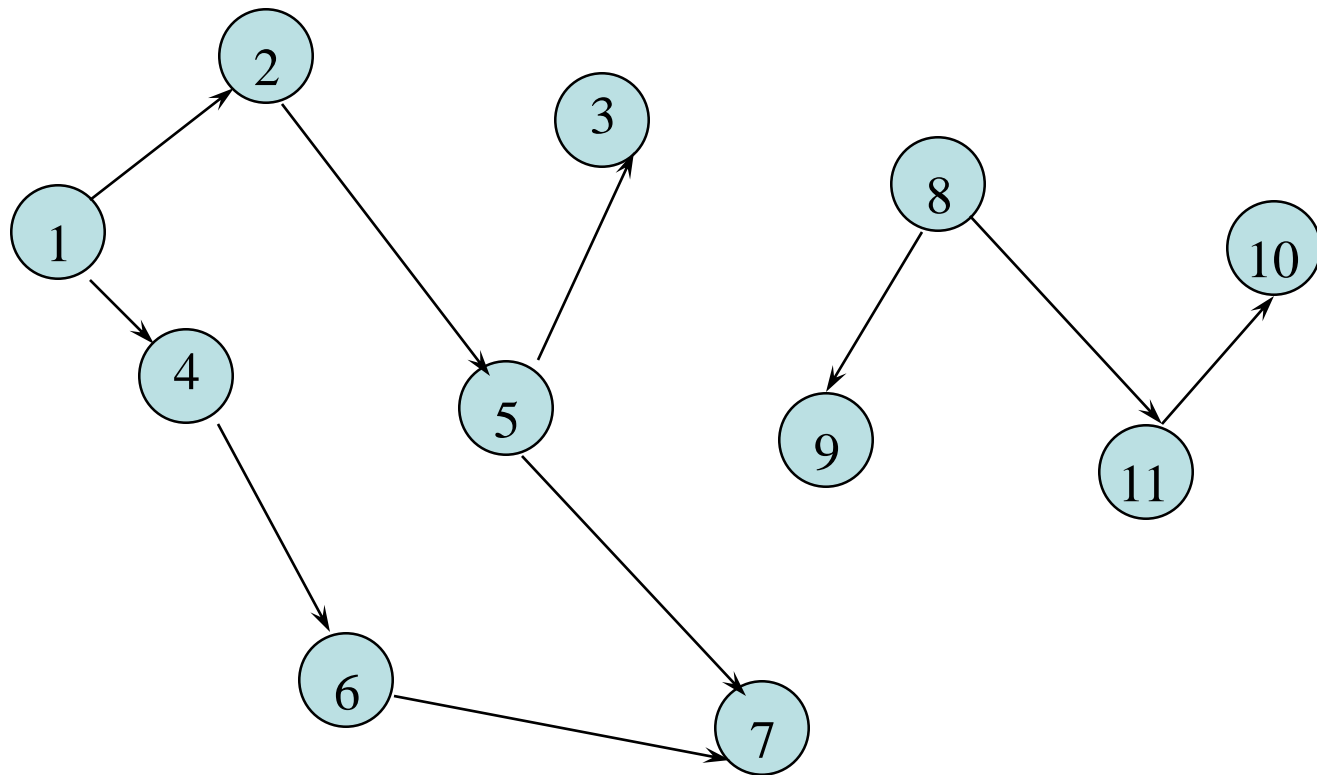
$u \text{ — } v$

- Чиглэлгүй граф \Rightarrow зүг чигийг заасан ирмэггүй.
- Чиглэлтэй граф \Rightarrow ирмэг бүр зүг чигийг заана.

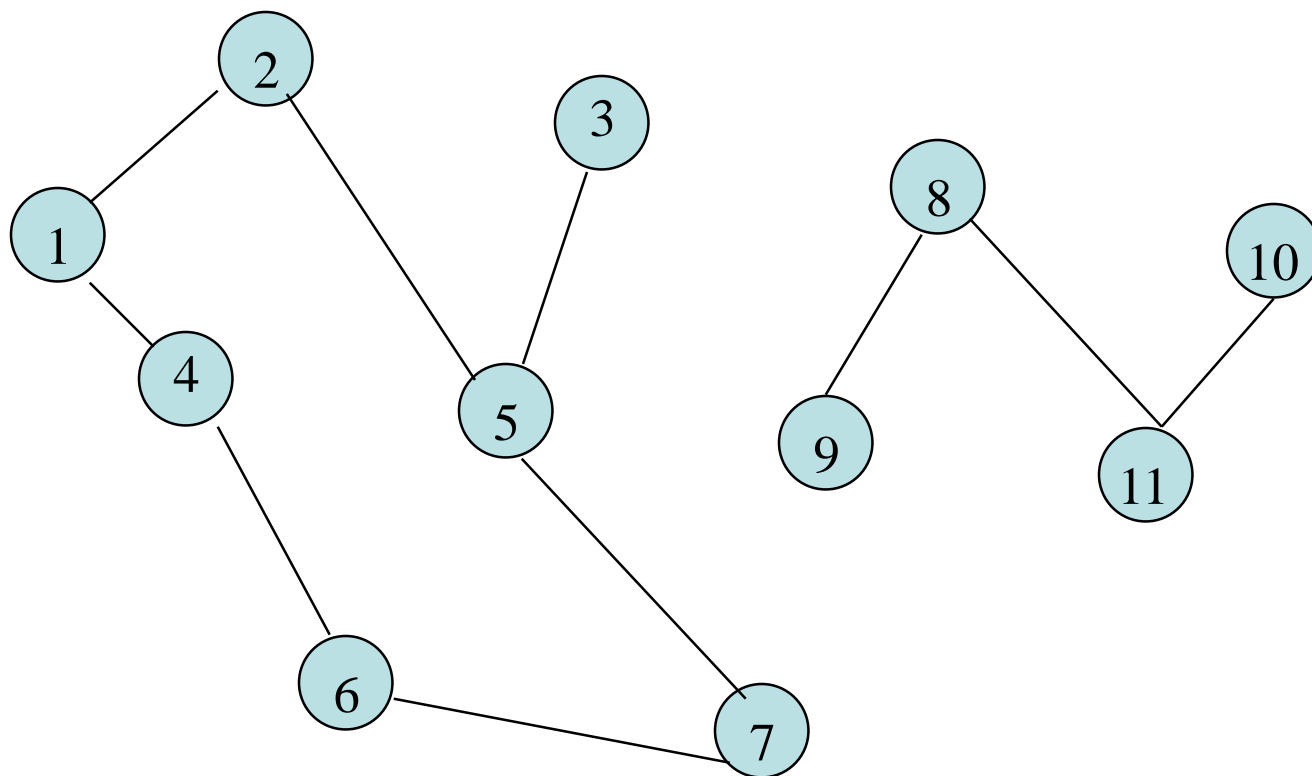
Чиглэлгүй граф



Чиглэлтэй граф (Digraph)

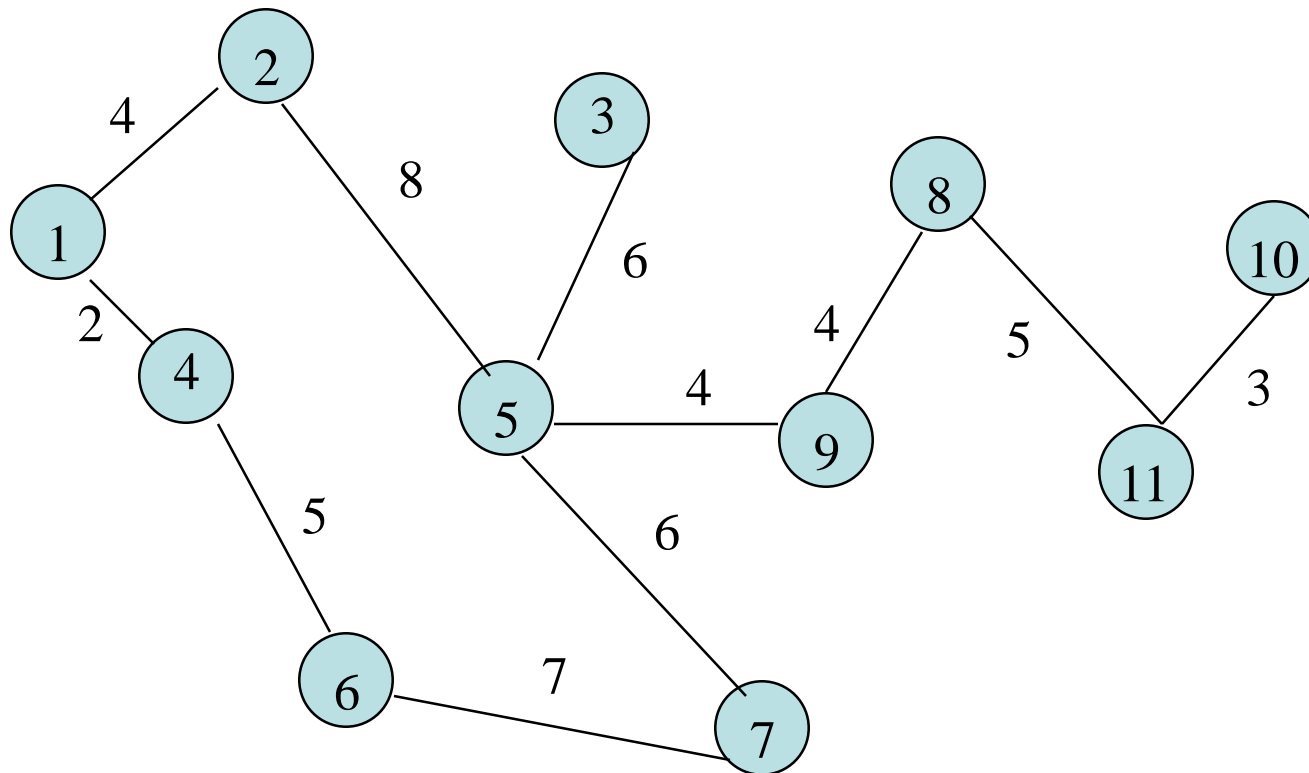


Хэрэглээ—Холбооний Сүлжээ



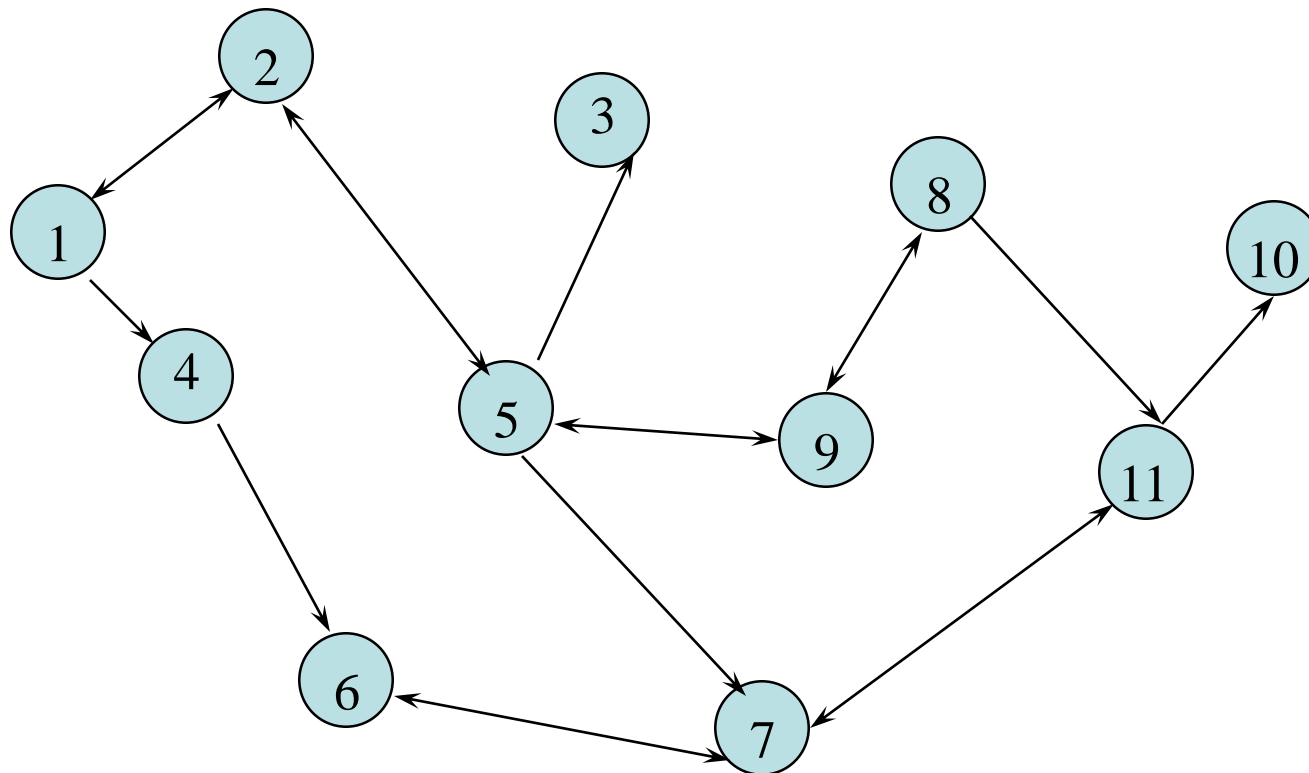
- Орой = хот, ирмэг = холбоос.

Явах зай/Хугацааны зураглал



- Орой = хот, ирмэгийн жин/ачаа = явах зай/хугацаа.

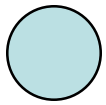
Гудамжны зураглал



- Зарим гудамж нэг чиглэлтэй.

Төгс чиглэлгүй граф

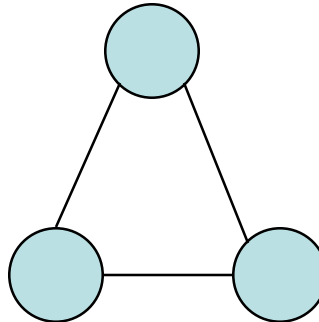
Байж болох бүх ирмэгтэй.



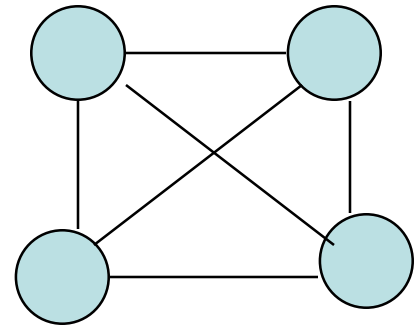
$n = 1$



$n = 2$



$n = 3$



$n = 4$

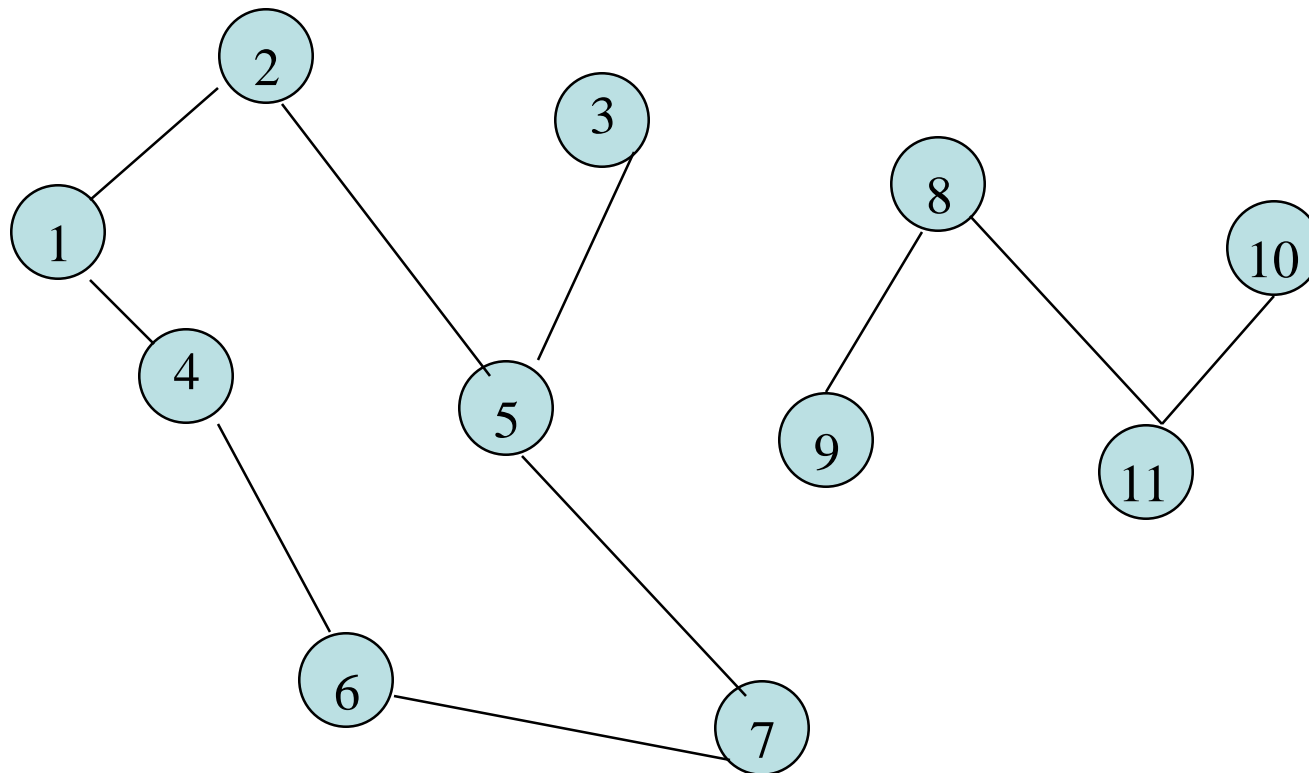
Ирмэгийн тоо—чиглэлгүй граф

- Ирмэг бүр (u,v) , $u \neq v$ хэлбэртэй
- n оройтой графын ийм хосын тоо $n(n-1)$.
- Нэгэнт (u,v) ирмэг (v,u) ирмэгтэй адил болохоор, төгс чиглэлгүй графын ирмэгийн тоо $n(n-1)/2$.
- Чиглэлгүй графын ирмэгийн тоо $\leq n(n-1)/2$.

Ирмэгийн тоо—Чиглэлтэй граф

- Ирмэг бүр (u,v) , $u \neq v$ хэлбэртэй
- n оройтой графын ийм хосын тоо $n(n-1)$.
- Нэгэнт (u,v) ирмэг (v,u) ирмэгтэй адилгүй тул төгс чиглэлтэй графын ирмэгийн тоо $n(n-1)$.
- Чиглэлтэй графын ирмэгийн тоо $\leq n(n-1)$.

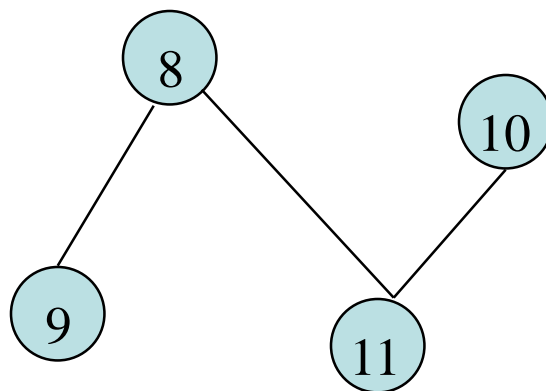
Оройн зэрэг(Vertex degree)



Орой руу явсан ирмэгийн тоо.

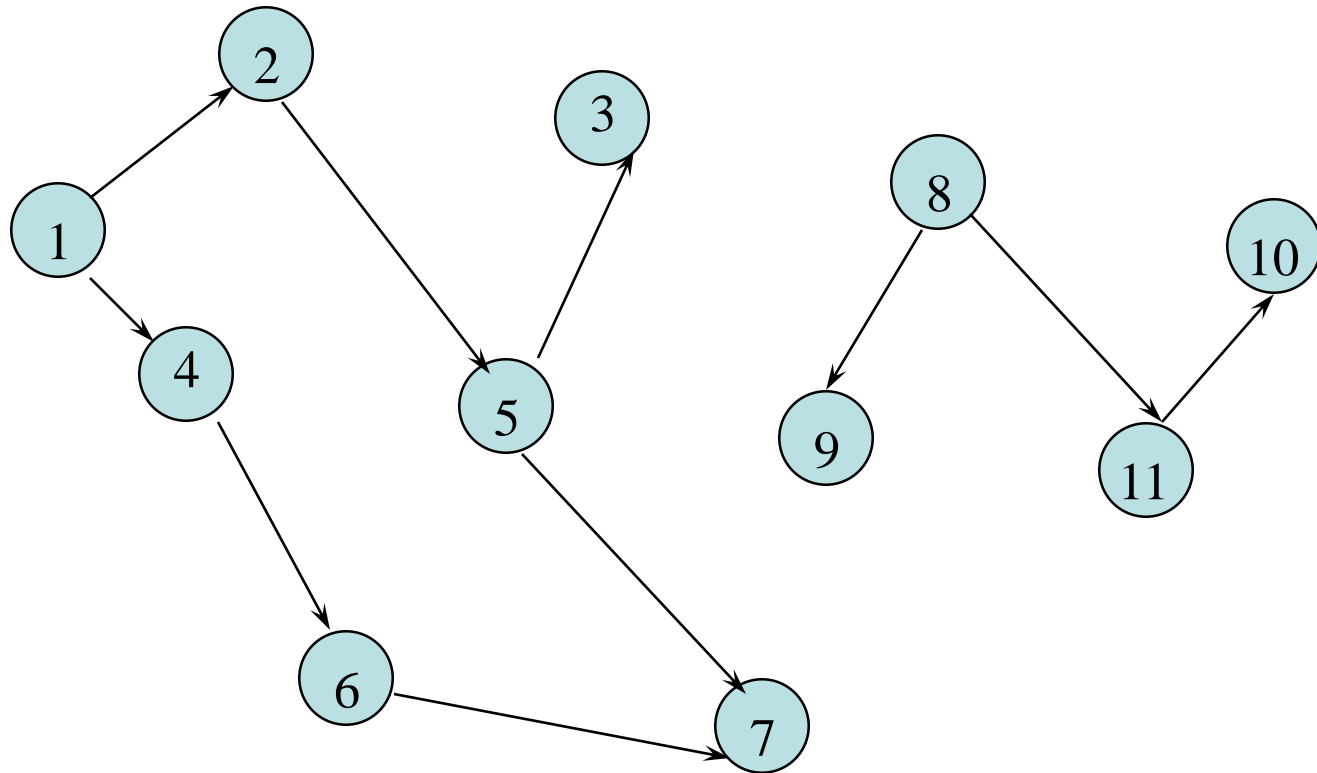
$\text{degree}(2) = 2$, $\text{degree}(5) = 3$, $\text{degree}(3) = 1$

Оройн зэргийн нийлбэр



Зэргийн нийлбэр = $2e$ (e ирмэгийн тоо)

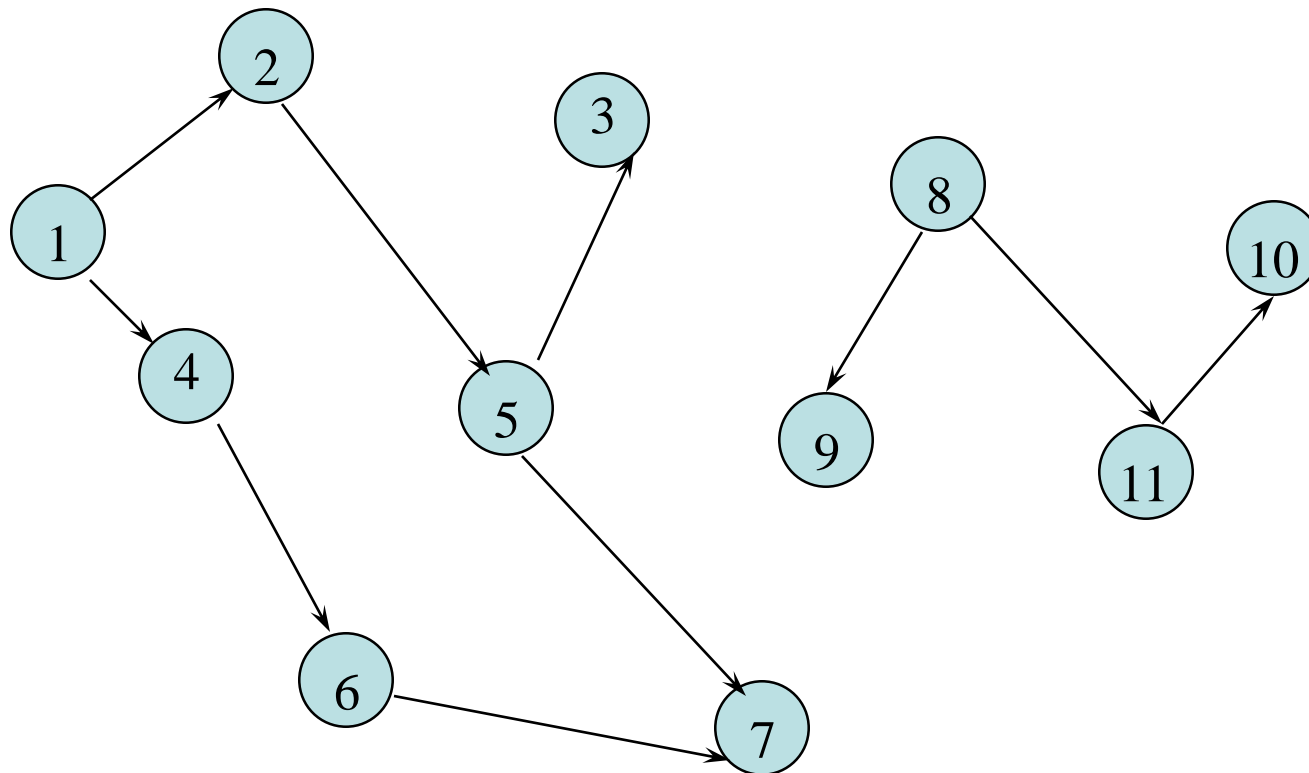
Оройн орох зэрэг(In-Degree)



in-degree – орсон ирмэгийн тоо

$\text{indegree}(2) = 1$, $\text{indegree}(8) = 0$

Оройн гарах зэрэг(Out-Degree)



out-degree - гарсан ирмэгийн тоо

$\text{outdegree}(2) = 1$, $\text{outdegree}(8) = 2$

Орох ба гарах зэргийн нийлбэр

Ирмэг бүр 1-г ямар нэг оройн орох зэрэгт, 1-г нөгөө оройн гарах зэрэгт нэмэрлэдэг

Орох зэргийн нийлбэр = гарах зэргийн нийлбэр
= e ,

үүнд e – digraph ирмэгийн тоо

Графын үйлдлүүд ба дүрслэл

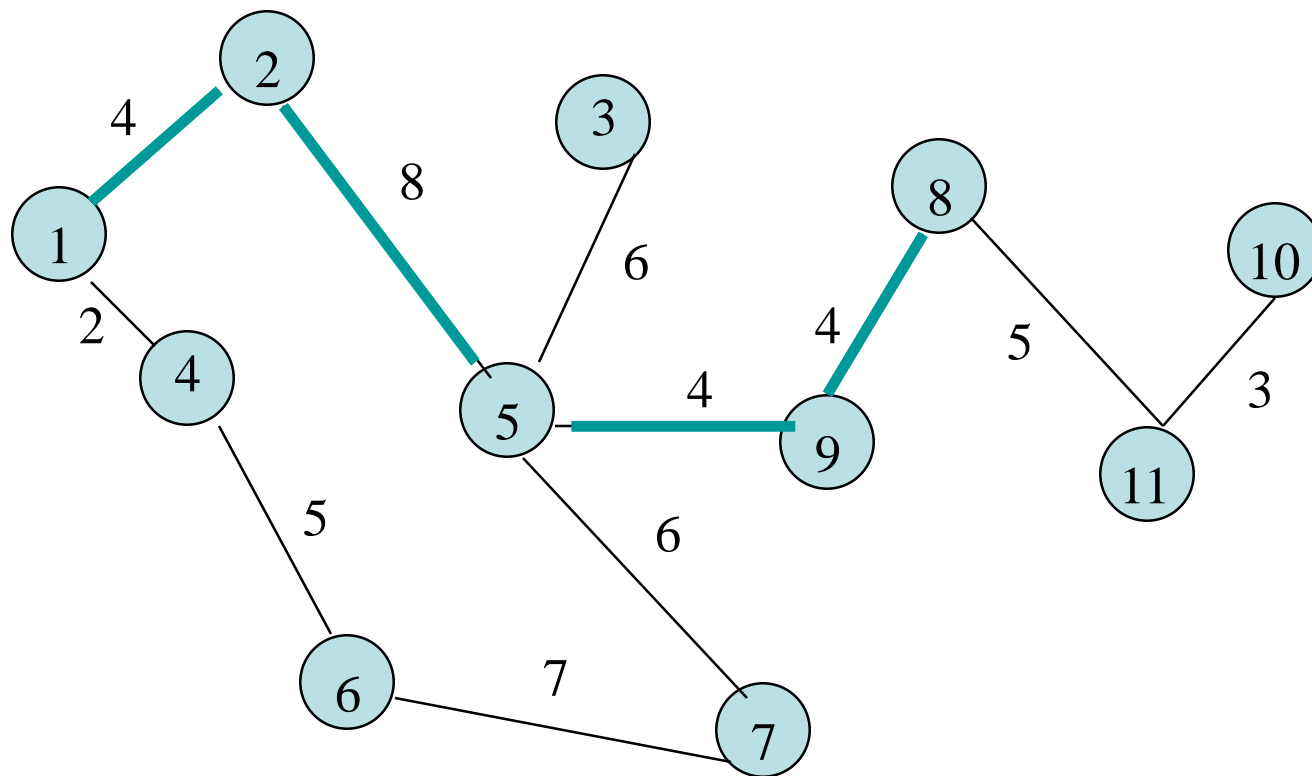


Графын зарим бодлогууд

- Замын бодлого.
- Холболтын бодлого.
- Бүрхэгч модны (Spanning tree) бодлого.

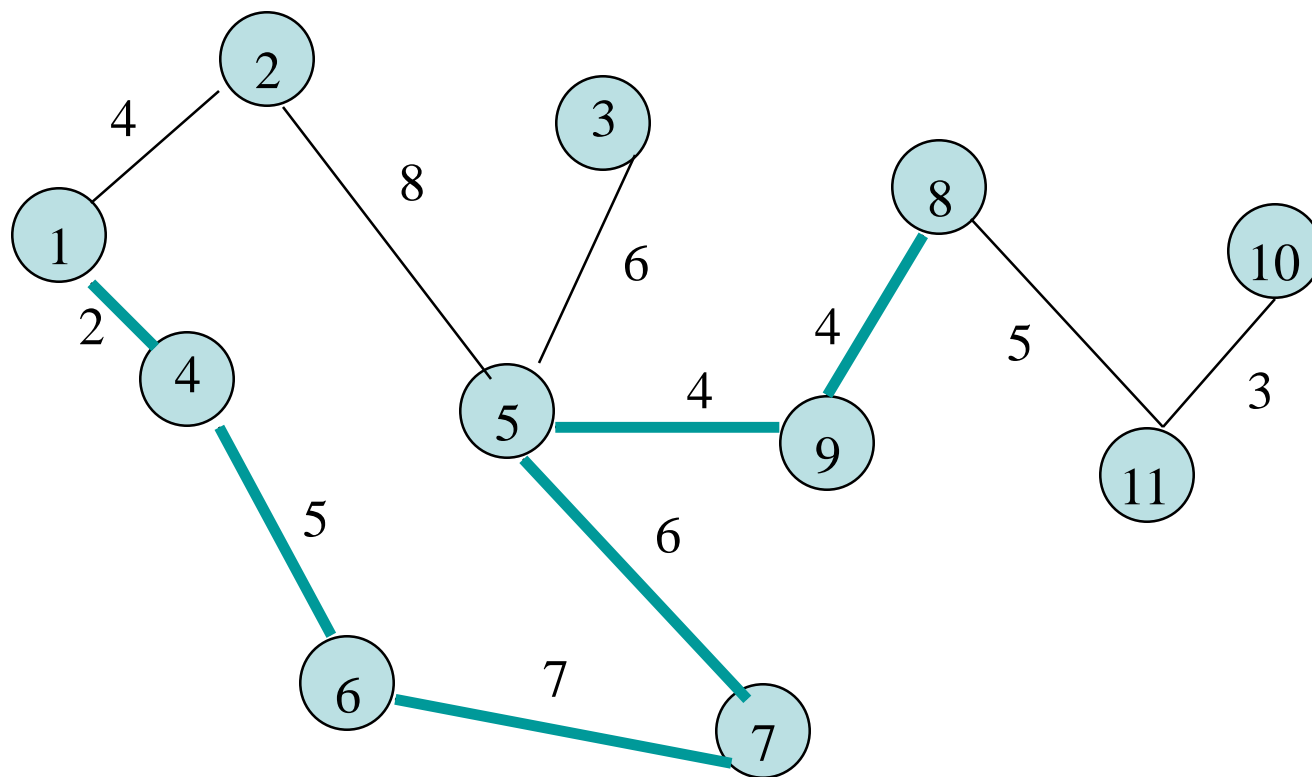
Зам олох

1 – 8 хүрэх зам



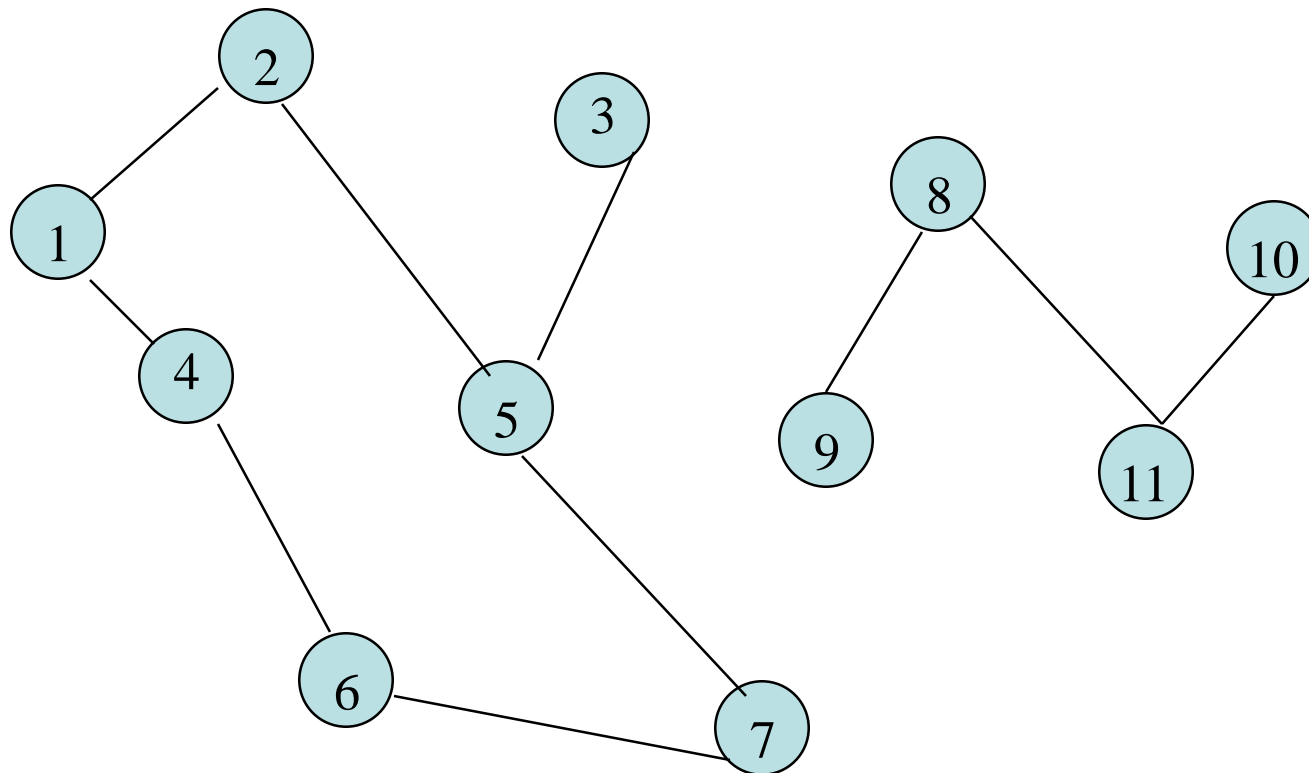
Замын урт 20.

1 – 8 хүрэх өөр зам



Замын урт 28.

Замгүй байх жишээ

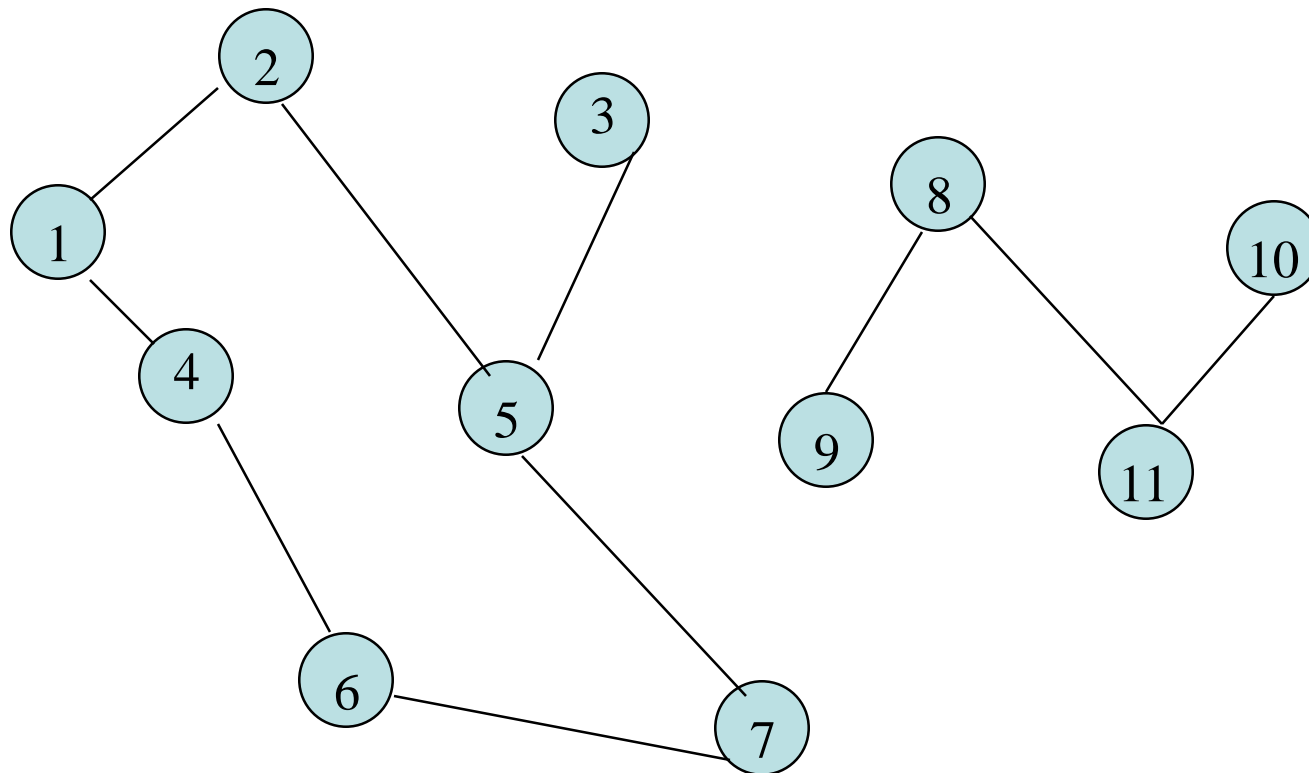


2 , 9 -ийн хооронд замгүй

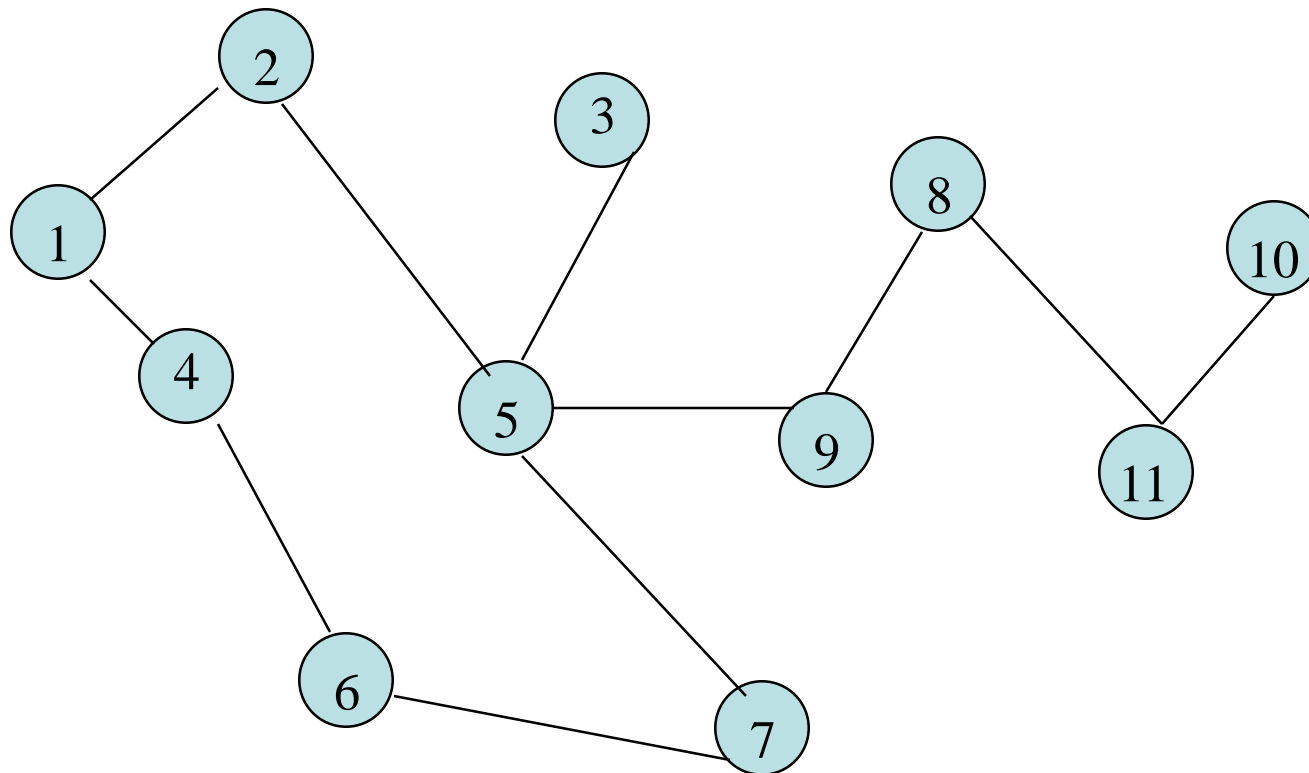
Холбогдсон граф

- Чиглэлгүй граф.
- Хос орой бүрийг хооронд замтай.

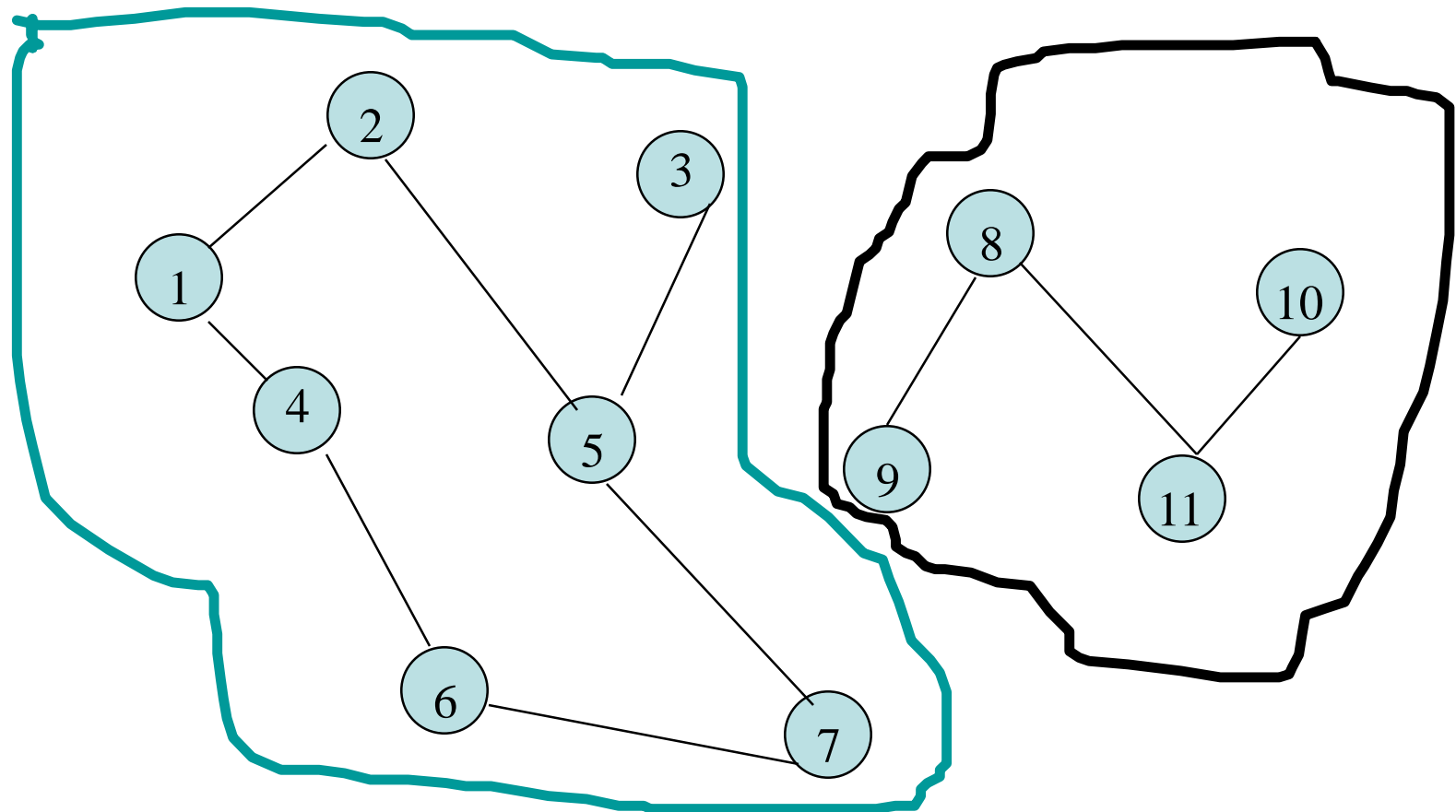
Холбогдоогүй графын жишээ



Холбогдсон графын жишээ



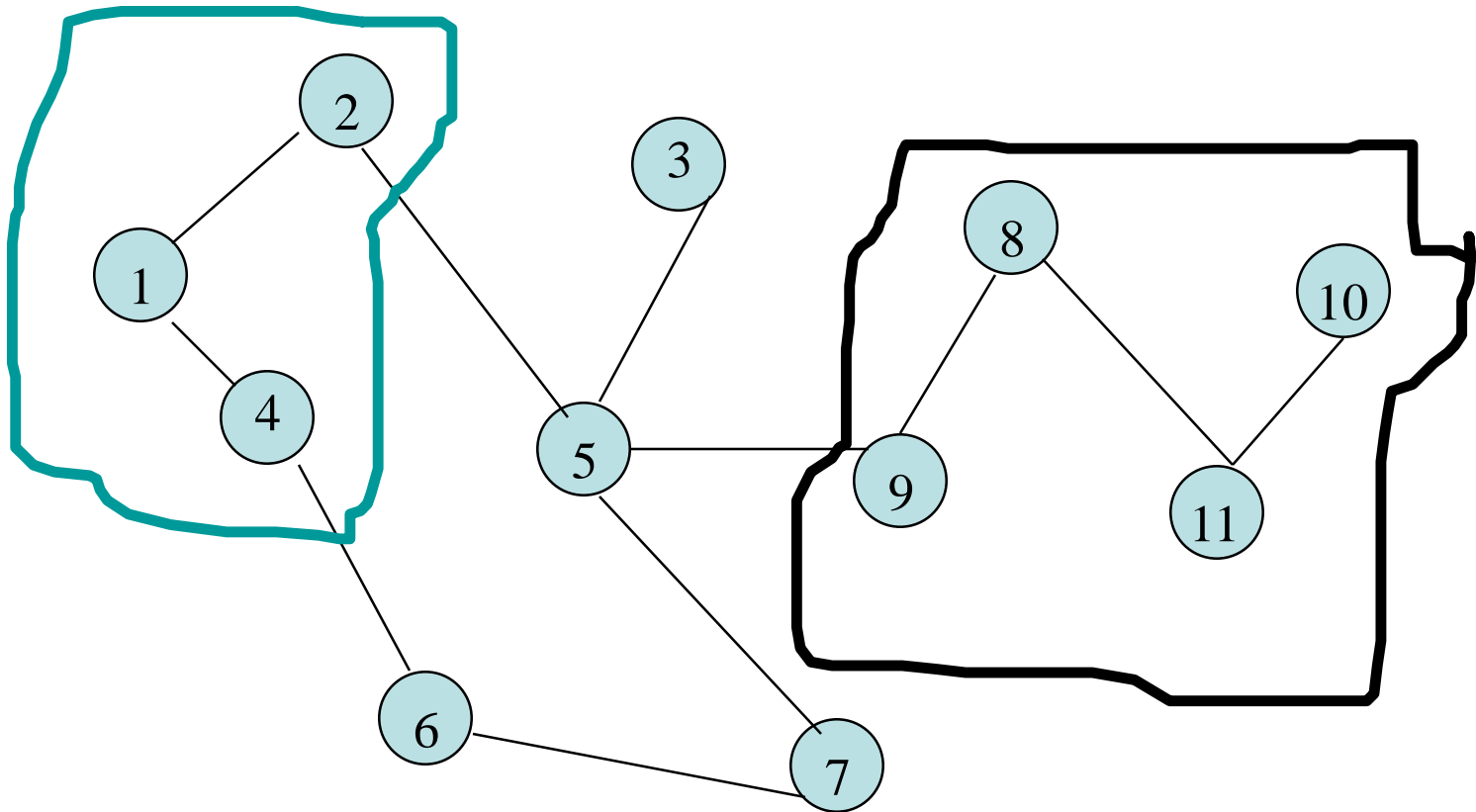
Холбогдсон бүрдүүлбэрүүд



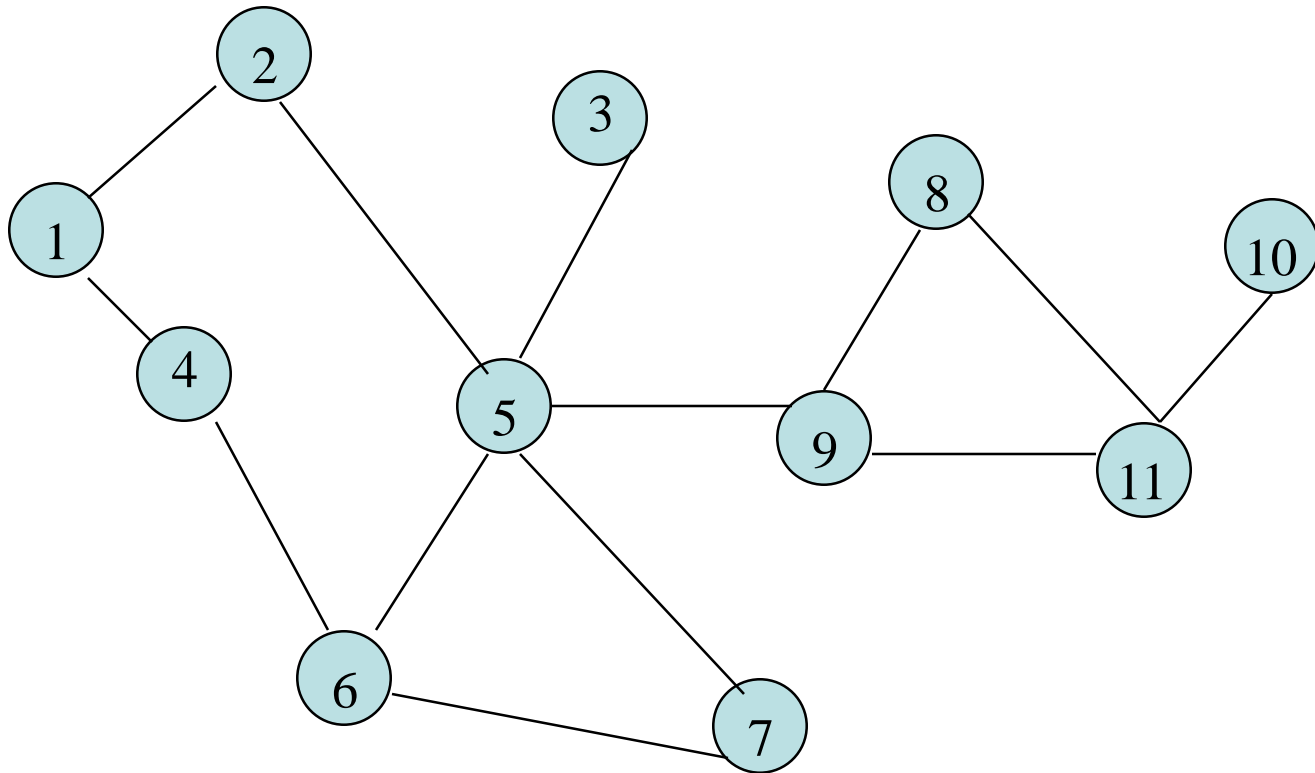
Холбогдсон бүрдүүлбэрүүд

- Холбогдсон дэд граф.
 - Оригинал граф дээр орой, ирмэг нэмж болохгүй, салангид байдлаа хадгалдаг.
- Холбогдсон граф яг 1 бүрдүүлбэртэй.

Бүрдүүлбэр биш



Холбооны сүлжээ

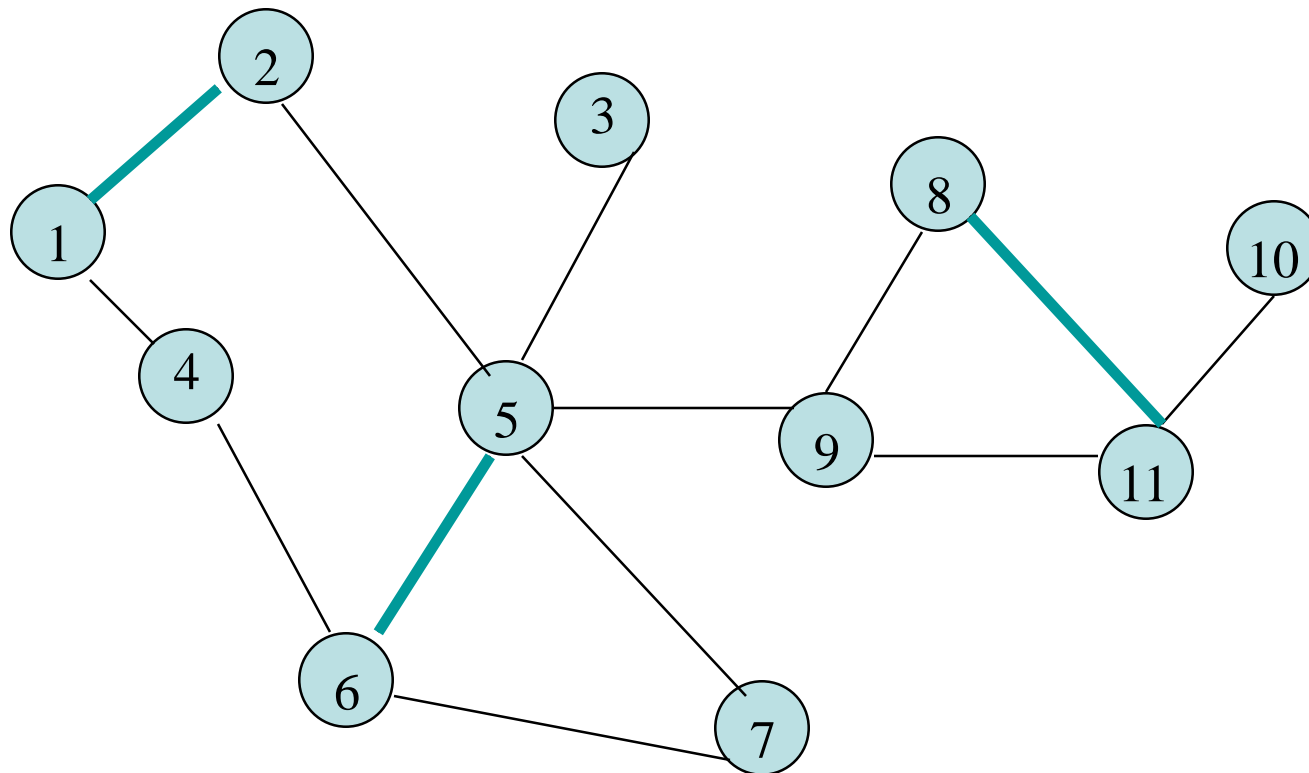


Ирмэг бүр байгуулж болох холбоос (ө.х., а боломжит холбоос).

Холбооны сүлжээний бодлогууд

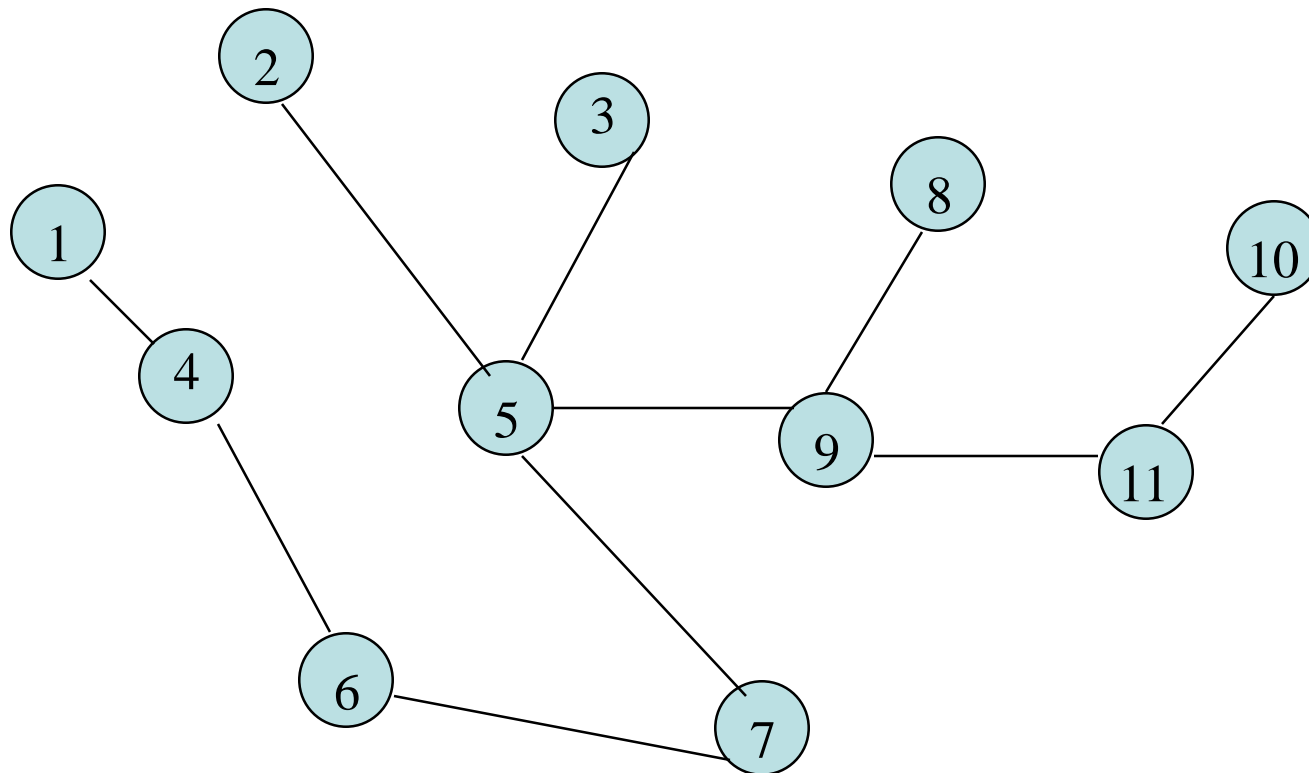
- Сүлжээ холбогдсон уу?
 - Дурын хоёр хот харьцаж чадах уу?
- Бүрдүүлбэрийг олох.
- Сүлжээ холбогдсон байхын тулд шаардлагатай хамгийн цөөн боломжит холбоосыг байгуулах.

Цикл ба холболт



Цагирагнаас ирмэг устгахад холболтонд
нөлөөлөхгүй.

Цикл ба холболт



Бүх орой, цикл үүсгэхгүй минимум тооны ирмэгтэй холбогдсон дэд граф.



Мод

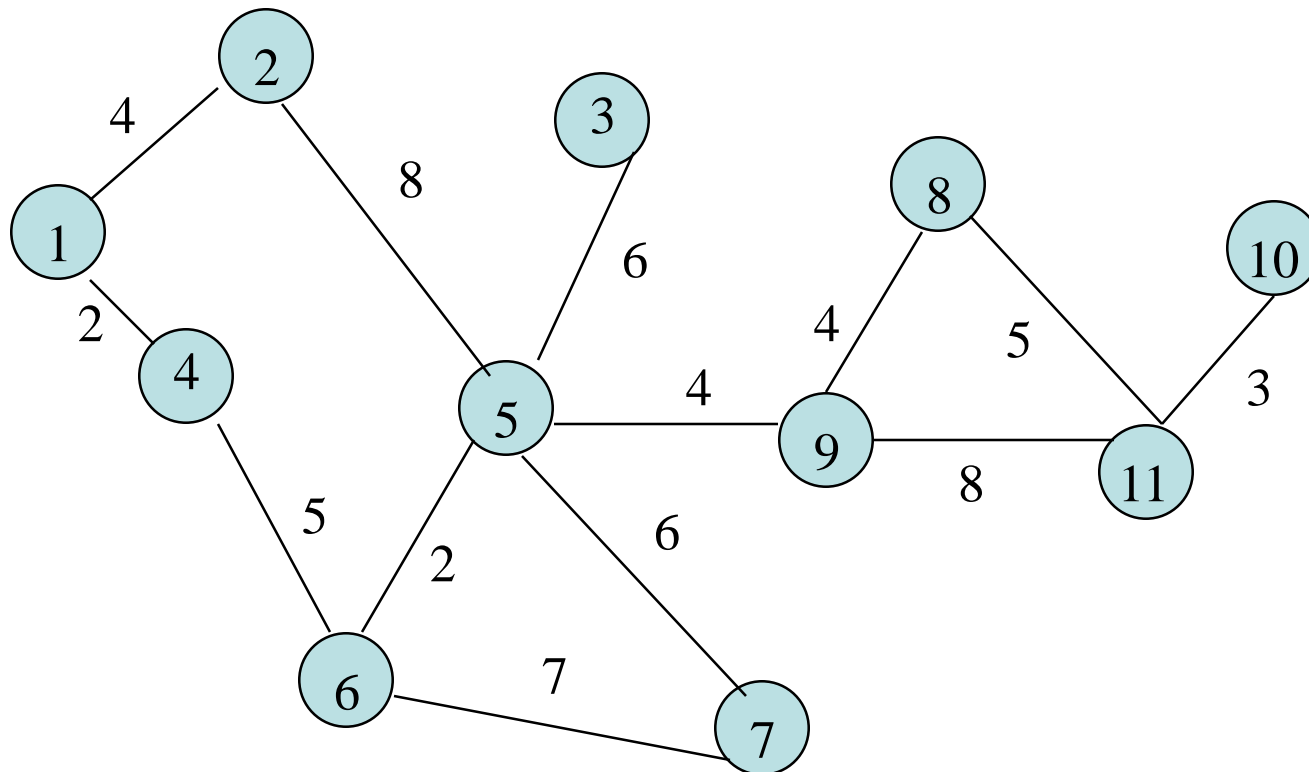


- Циклгүй, холбогдсон граф.
- $n-1$ ирмэгтэй, n оройтой холбогдсон граф.

Бүрхэгч мод

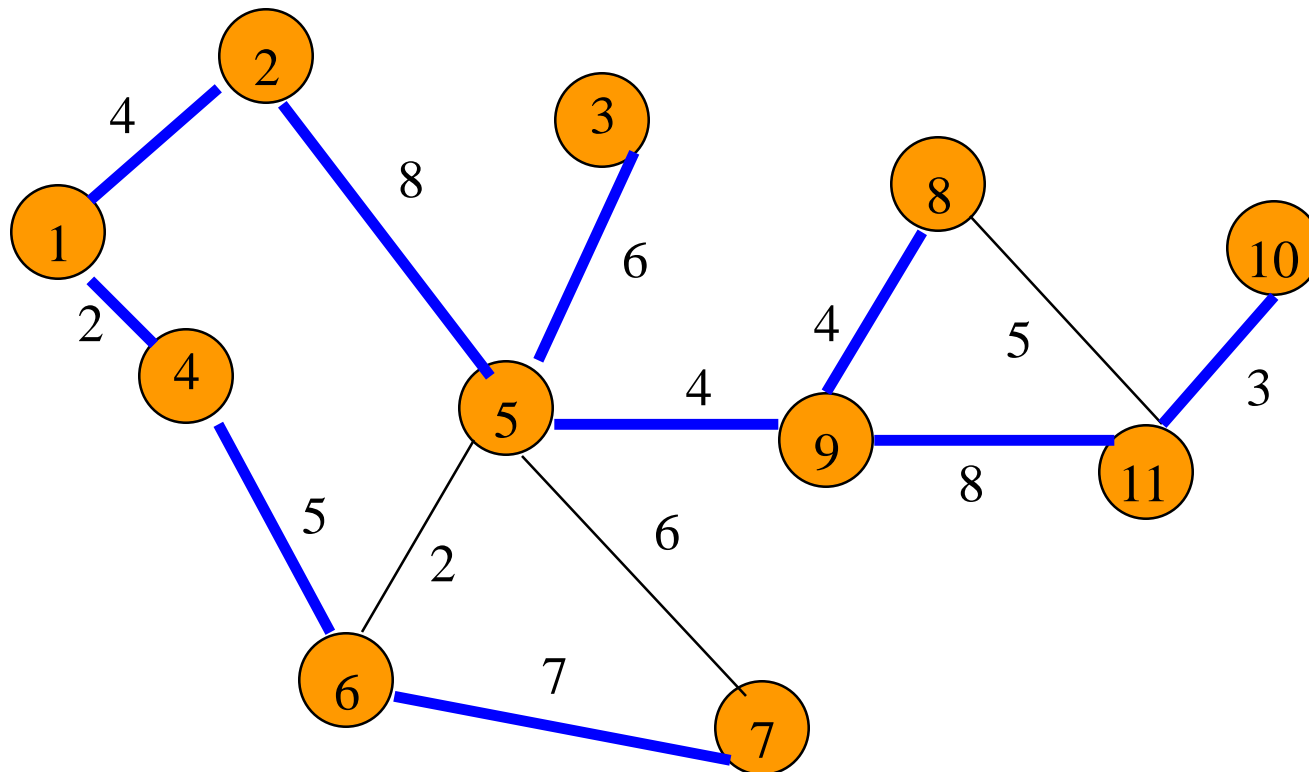
- Оригинал графын бүх оройг агуулсан дэд граф.
- Дэд граф нь мод.
 - Хэрвээ оригинал граф n оройтой бол, бүрхэлтийн мод n орой, $n-1$ ирмэгтэй.

Min өртөгтэй бүрхэгч мод



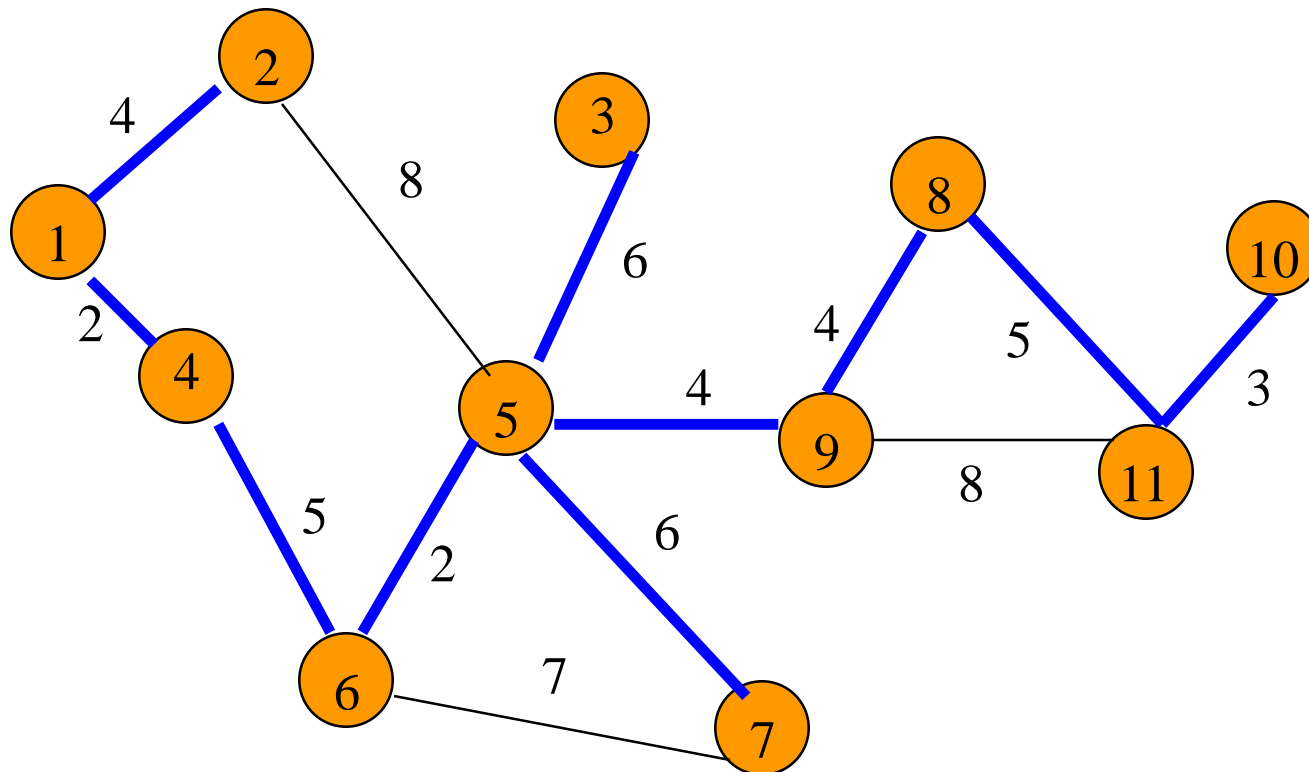
- Модны өртөг/зардал нь ирмэгүүдийн жин/өртгийн нийлбэр байна.

Бүрхэгч мод



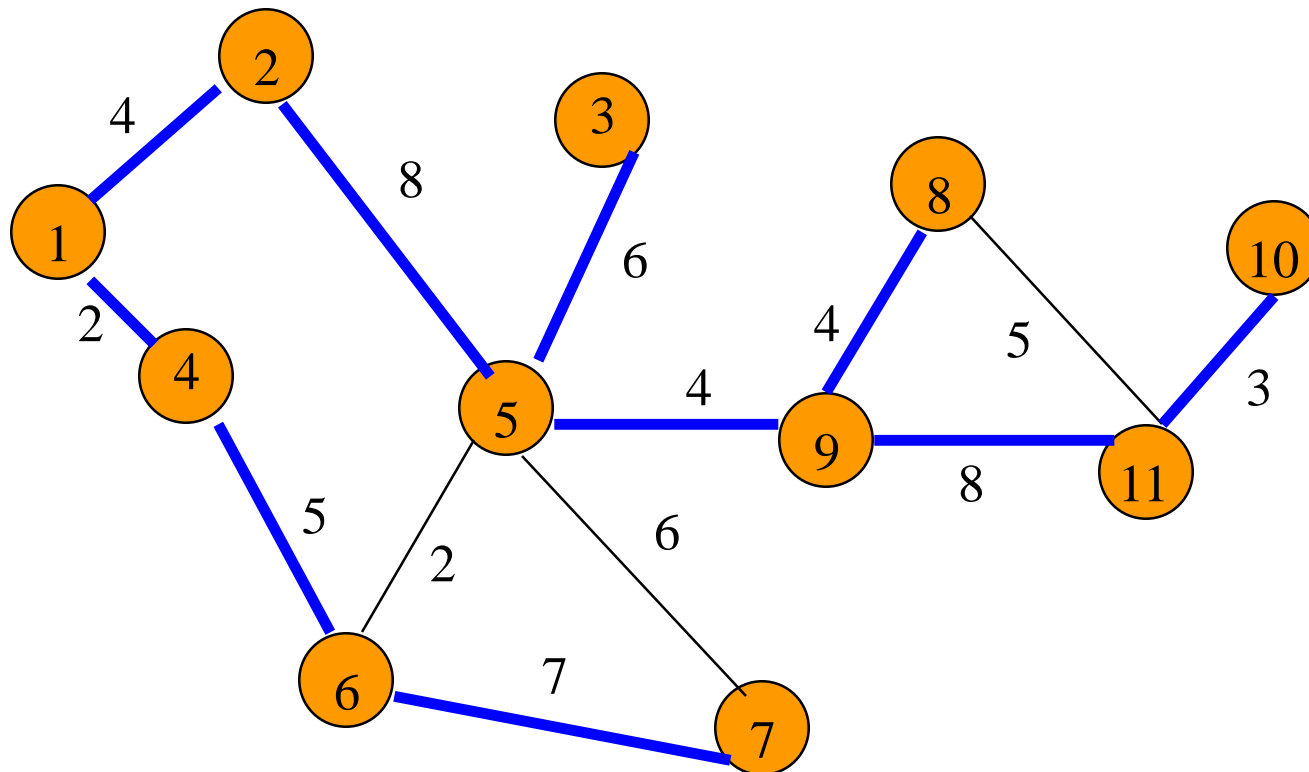
Бүрхэгч модны өртөг = 51.

Min өртгийн Бүрхэгч мод



Бүрхэгч модны өртөг = 41.

Утасгүй Дамжуулалтын мод



Төв = 1, жин = шаардлагатай чадал.

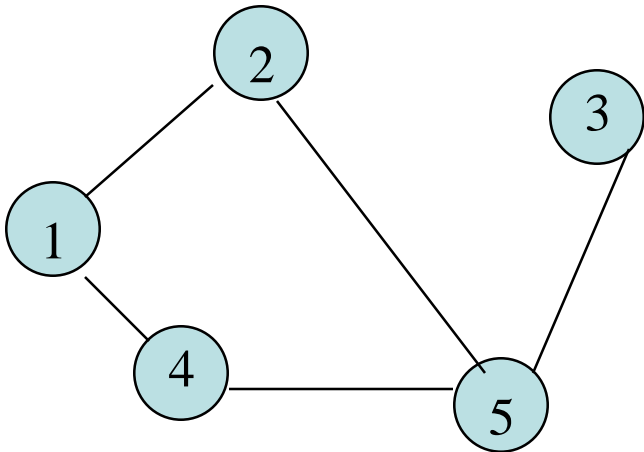
Зардал = $4 + 8 + 5 + 6 + 7 + 8 + 3 = 41$.

Графын дүрслэл

- Холболтын матриц(Adjacency matrix)
- Холболтын жагсаалт
 - Холбоост холболтын жагсаалт
 - Массивт холболтын жагсаалт

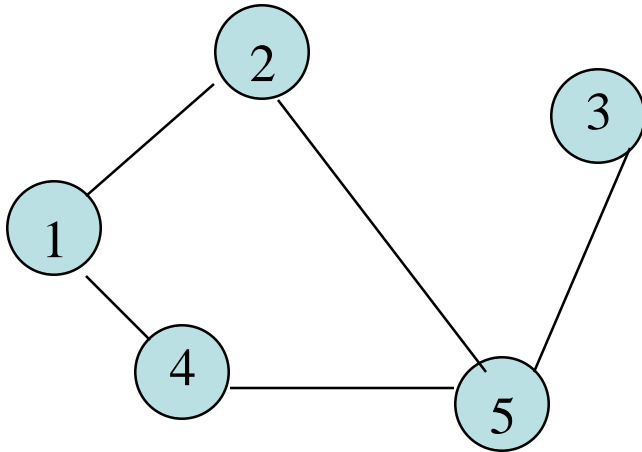
Холболтын матриц

- 0/1 $n \times n$ матриц, үүнд n = оройн тоо
- $A(i,j) = 1$ хэрвээ (i,j) гэсэн ирмэгтэй бол



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

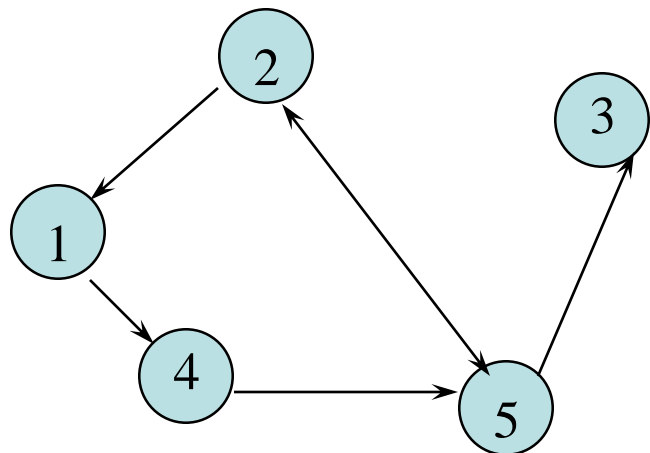
Холболтын матрицын шинж



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

- Диагональ 0.
- Чиглэлгүй графын холболтын матриц симметр.
 - $A(i,j) = A(j,i)$ бүх i, j –ийн хувьд

Холболтын матриц (Digraph)



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

- Диагональ 0.
- Чиглэлтэй графын холболтын матриц симметр бус.

Холболтын матриц

- n^2 бит орон зай
- Чиглэлгүй графын хувьд, дээд, доод гурвалжны аль нэг (диагональ орохгүй).
 - $(n-1)n/2$ бит
- $O(n)$ - оройн зэрэг, өгөгдсөн оройтой хөршлөх оройнуудыг олоход.

Холболтын жагсаалт

- i оройн холболтын жагсаалт нь тухайн оройтой хөршлөх оройнуудын шугаман жагсаалт байна.
- n холболтын жагсаалтын массив.

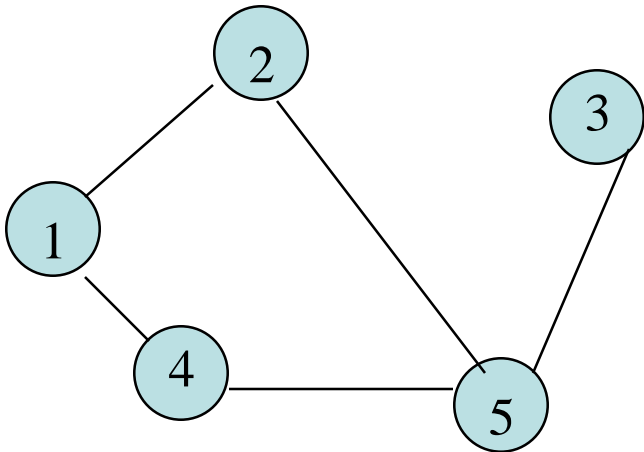
$$aList[1] = (2,4)$$

$$aList[2] = (1,5)$$

$$aList[3] = (5)$$

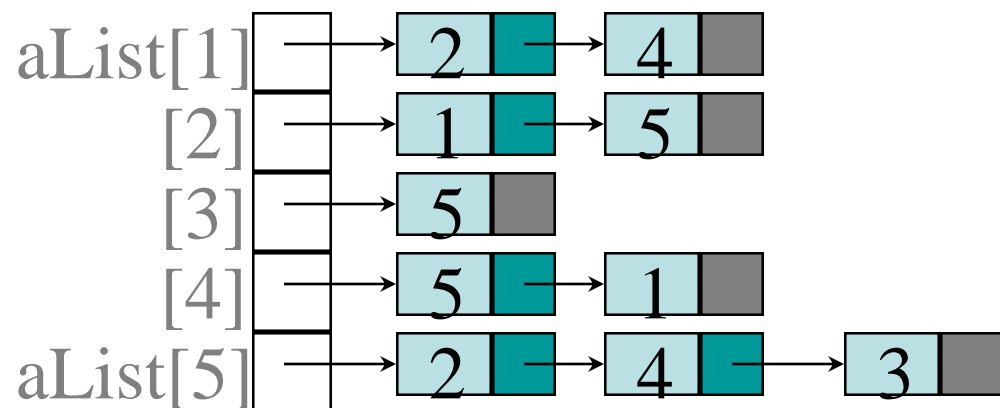
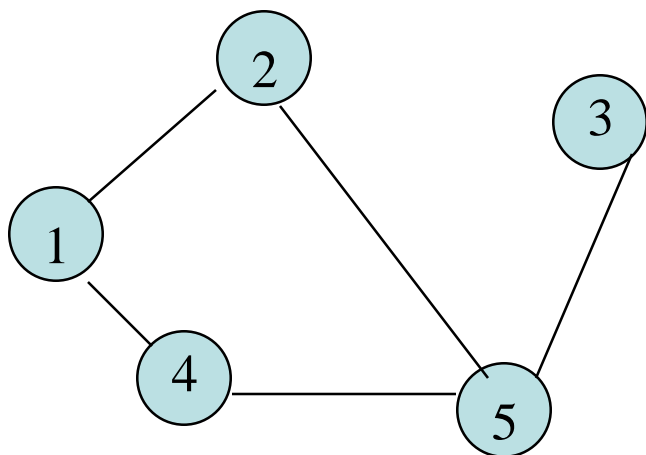
$$aList[4] = (5,1)$$

$$aList[5] = (2,4,3)$$



Холбоост холболтын жагсаалт

- Холболтын жагсаалт бүр гинж болно.



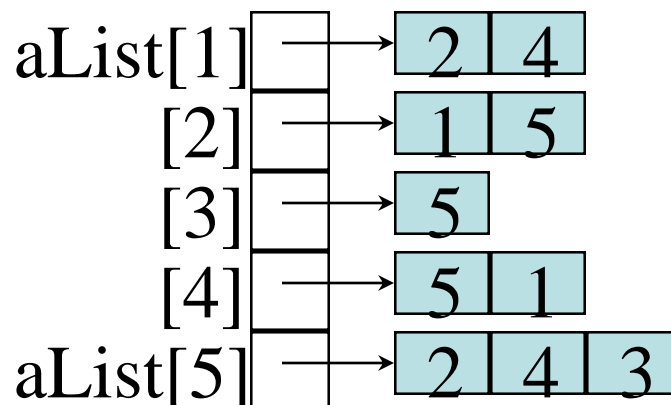
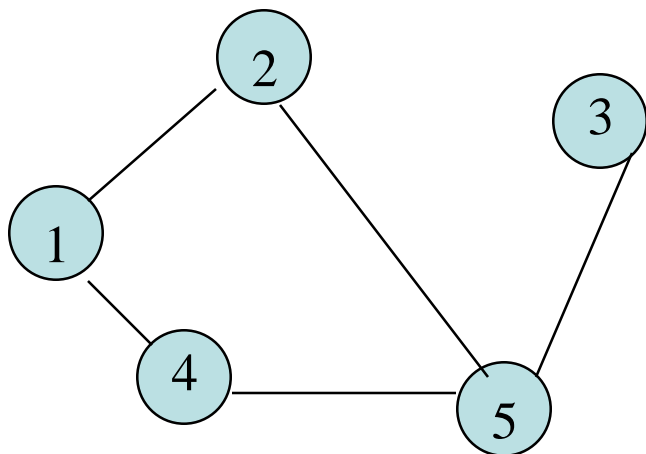
Массивын урт = n

Гинжин зангилааны тоо = $2e$ (чиглэлгүй граф)

Гинжин зангилааны тоо = e (digraph)

Массивт холболтын жагсаалт

- Холболтын жагсаалт бүр массив.



Массивын урт = n

Жагсаалтын элементийн тоо = $2e$ (чиглэлгүй граф)

Жагсаалтын элементийн тоо = e (digraph)

Жин/ачаатай граф

- Өртөгт холболтын матриц.
 - $C(i,j) = (i,j)$ ирмэгийн өртөг
- Холболтын жагсаалт \Rightarrow жагсаалтын элемент бүр хос (хөрш орой, ирмэгийн жин)

Шаардлагатай Java класс

- Графын дүрслэл
 - Холболтын матриц
 - Холболтын жагсаалт
 - Холбоост холболтын жагсаалт
 - Массивт холболтын жагсаалт
 - 3 дүрслэл
- Графын төрөл
 - Чиглэлтэй, чиглэлгүй.
 - Жинтэй, жингүй.
 - $2 \times 2 = 4$ графын төрөл
- $3 \times 4 = 12$ Java класс

Хийсвэр класс Graph

```
package dataStructures;  
import java.util.*;  
public abstract class Graph  
{  
    // Хийсвэр өгөгдлийн төрлийн аргууд энд бичигдэнэ  
  
    // i оройн iterator арга  
    public abstract Iterator iterator(int i);  
  
    // хэрэгжүүлэлтээс хамааралгүй аргуудын хэрэгжүүлэлт  
    энд бичигдэнэ  
}
```

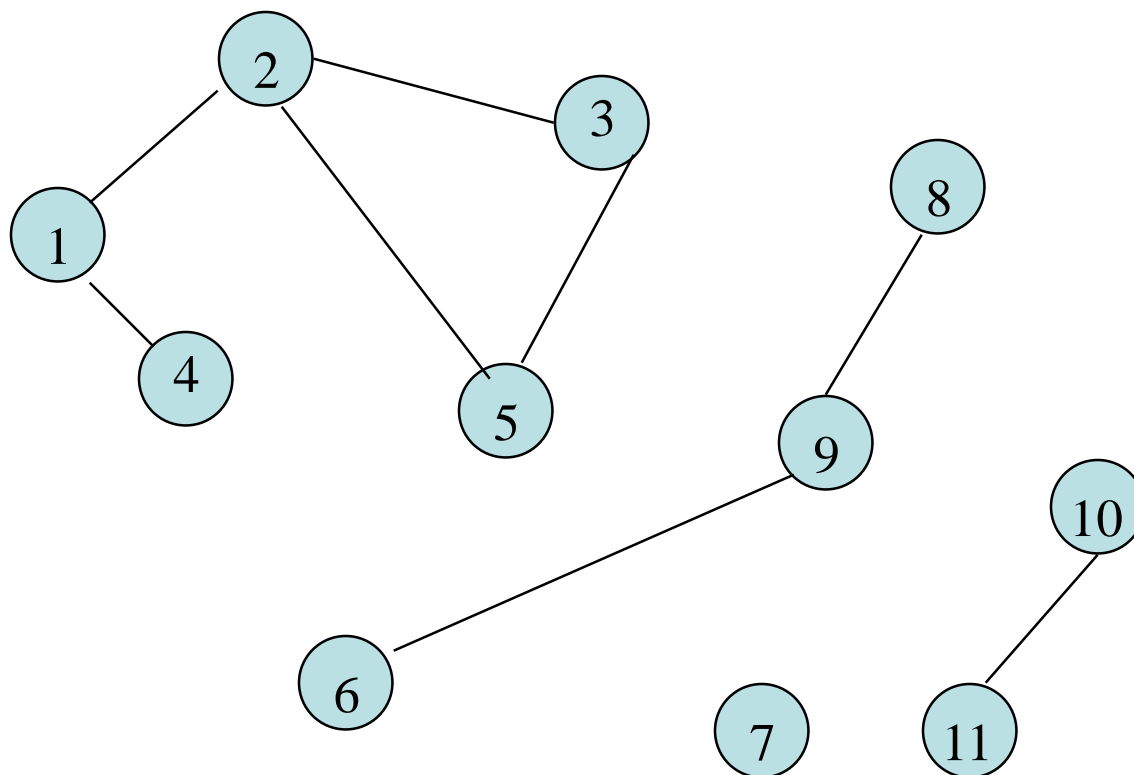
Graph –н хийсвэр аргууд

// ADT methods

```
public abstract int vertices();  
public abstract int edges();  
public abstract boolean existsEdge(int i, int j);  
public abstract void putEdge(Object theEdge);  
public abstract void removeEdge(int i, int j);  
public abstract int degree(int i);  
public abstract int inDegree(int i);  
public abstract int outDegree(int i);
```

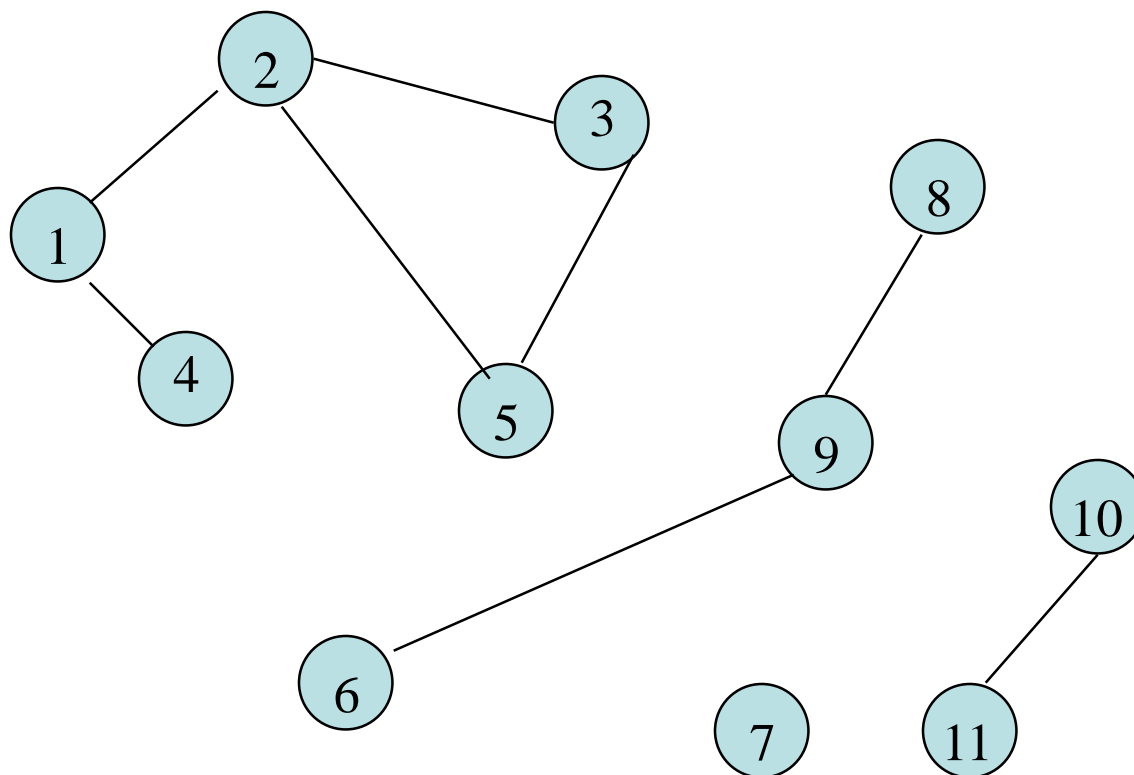

Графын хайлтын аргууд

- Хэрвээ v -с u –н хооронд зам байгаа бол v оройгоос u оройд **хүрч болно**



Графын хайлтын аргууд

- Хайлтын арга өгөгдсөн v оройгоос эхэлж v оройгоос хүрч болох бүх оройгоор зочлож/хаяглаж/тэмдэглэж хайна.



Графын хайлтын аргууд

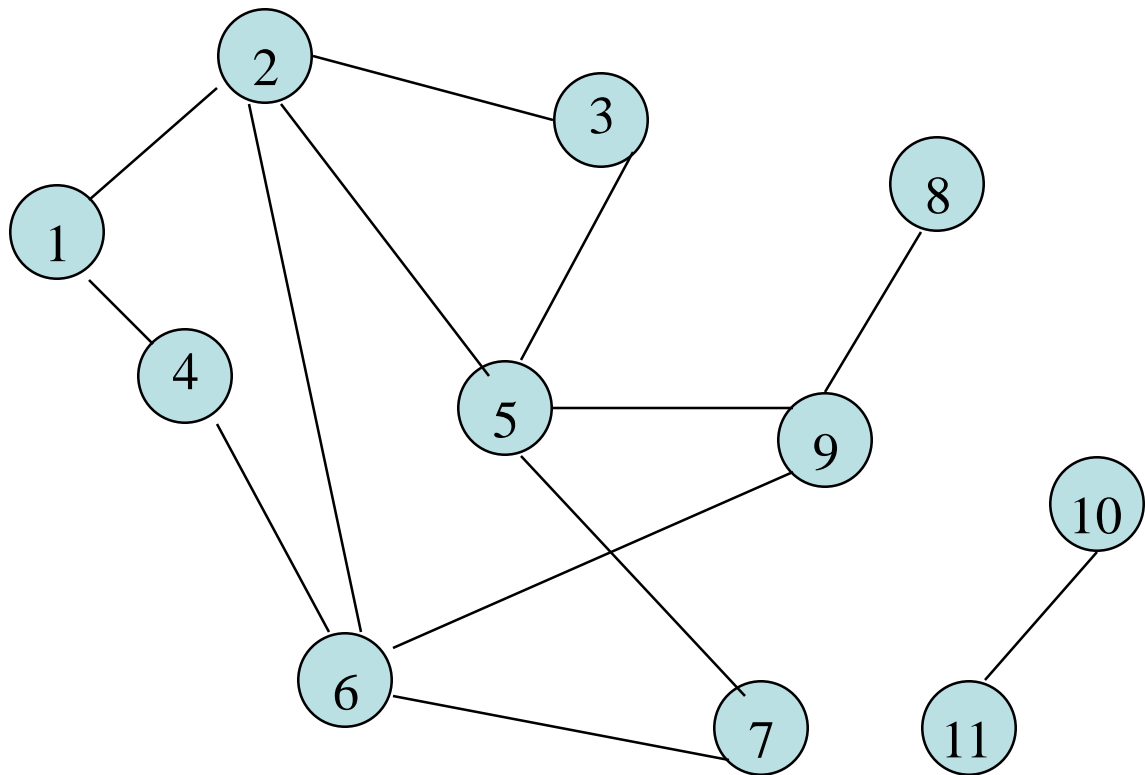
- Ихэнх графын бодлогуудыг хайлтын аргаар шийдвэрлэдэг.
 - Нэг оройгоос нөгөө оройд хүрэх зам.
 - Граф холбогдсон эсэх?
 - Бүрхсэн модыг олох.
 - Г.м.
- Түгээмэл ашигладаг хайлтын аргууд:
 - Түвшнээр-Эхэлж хайх.(Depth-First-Search) түвшний хайлт
 - Гүнээр-Эхэлж хайх.(Breadth-First-Search) гүний хайлт

Түвшний хайлт

(Breadth-First Search)

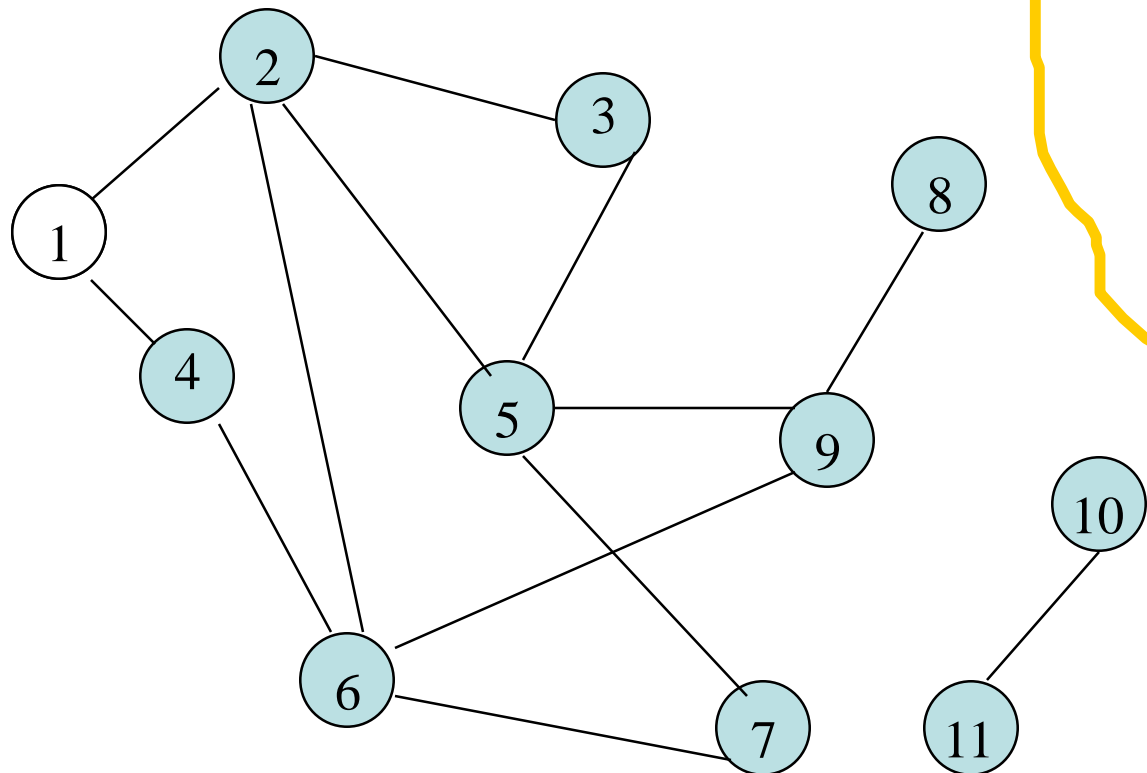
- Энхний оройд зочлоод түүнийг FIFO дараалалд хийнэ.
- Дараалаас оройг устгаж, уг оройн зочлоогүй хөрш оройнуудаар зочлох, шинэ зочилсон оройнуудаа дараалалд нэмэх үйлдлийг давтан гүйцэтгэнэ.

Түвшний хайлтын жишээ



Хайлт эхлэх орой - 1.

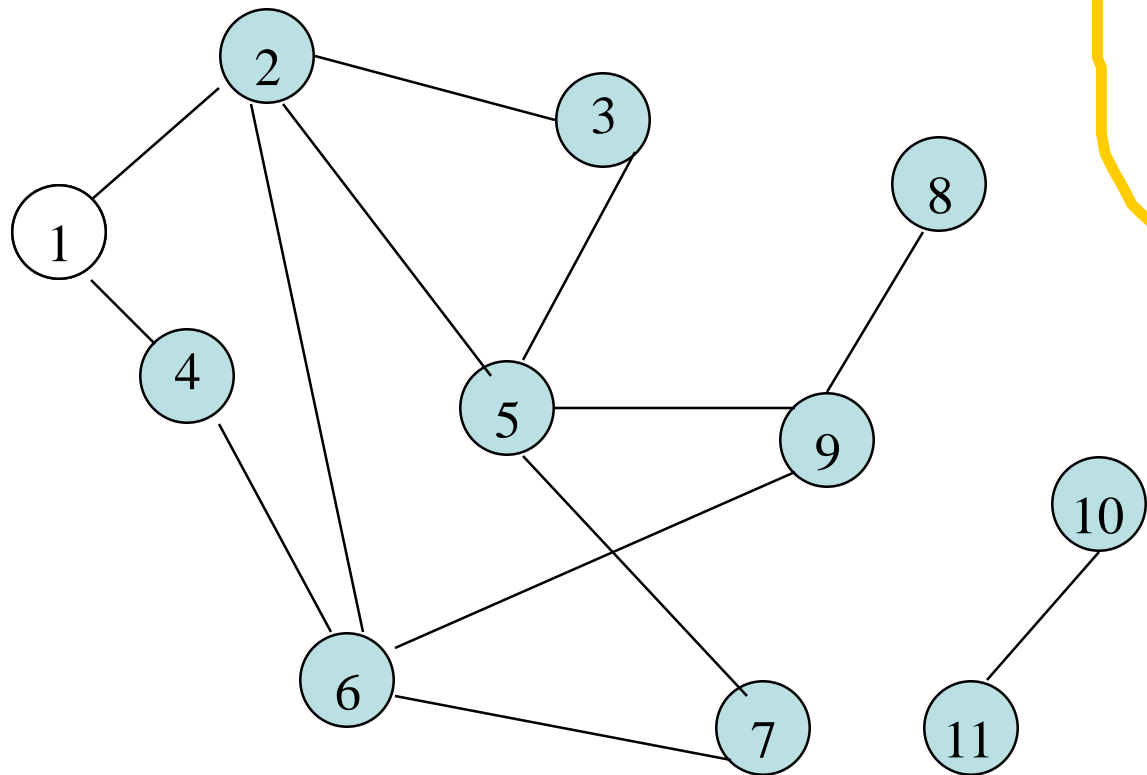
Түвшний хайлтын жишээ



FIFO дараалал
1

Эхний оройд зочилж/тэмдэглэж/хаяглаад, FIFO дараалалд хийнэ.

Түвшний хайлтын жишээ



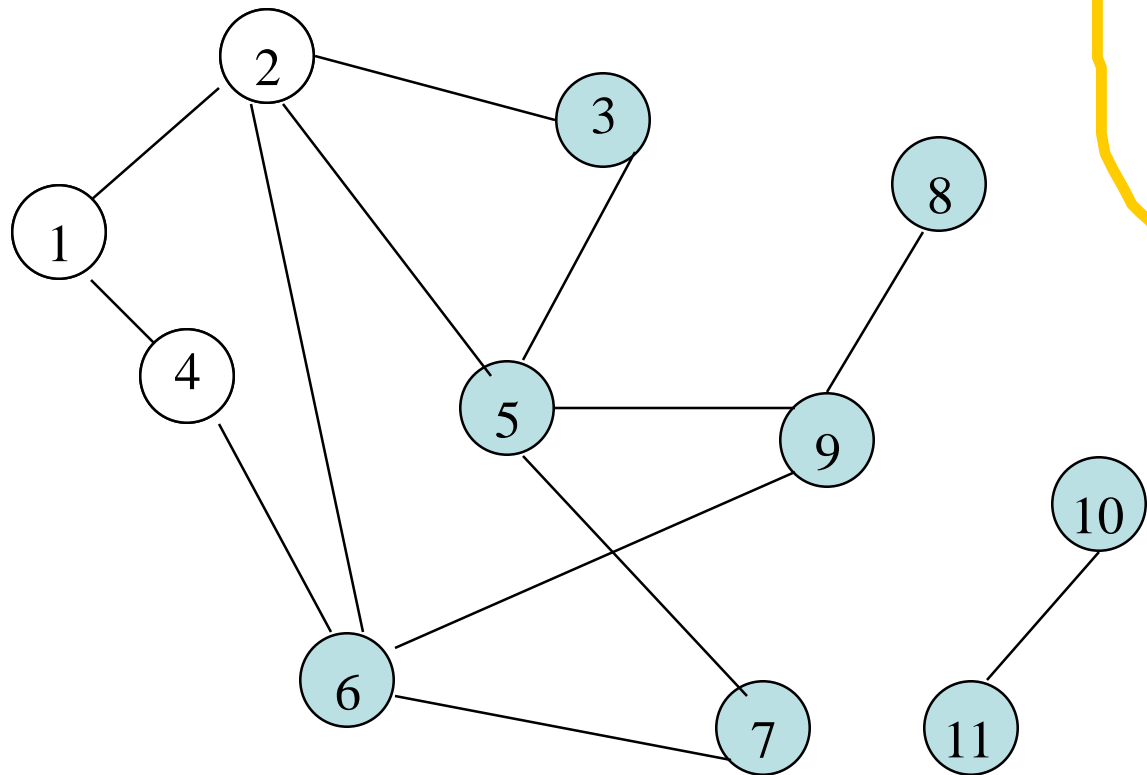
FIFO дараалал

1

Q -с 1 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



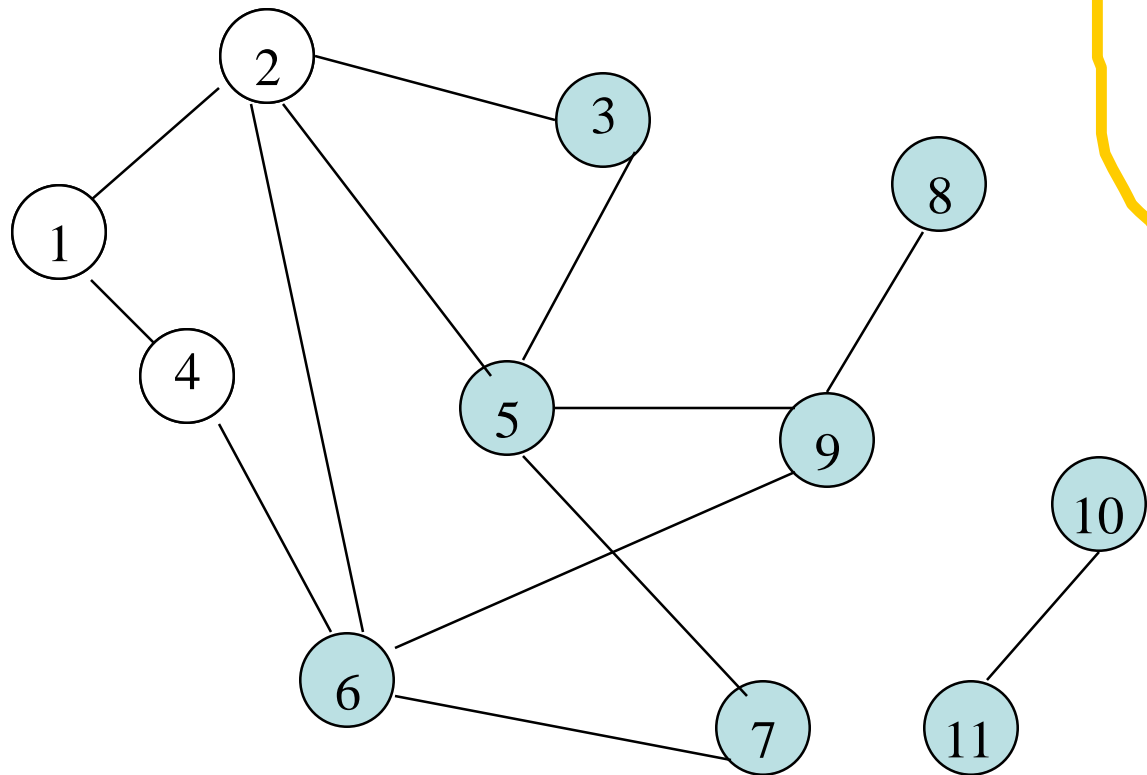
FIFO дараалал

2 4

Q -с 1 –г устгаж, айлчлаагүй хөршүүдээр нь айлчилж

Q -д хийнэ.

Түвшний хайлтын жишээ



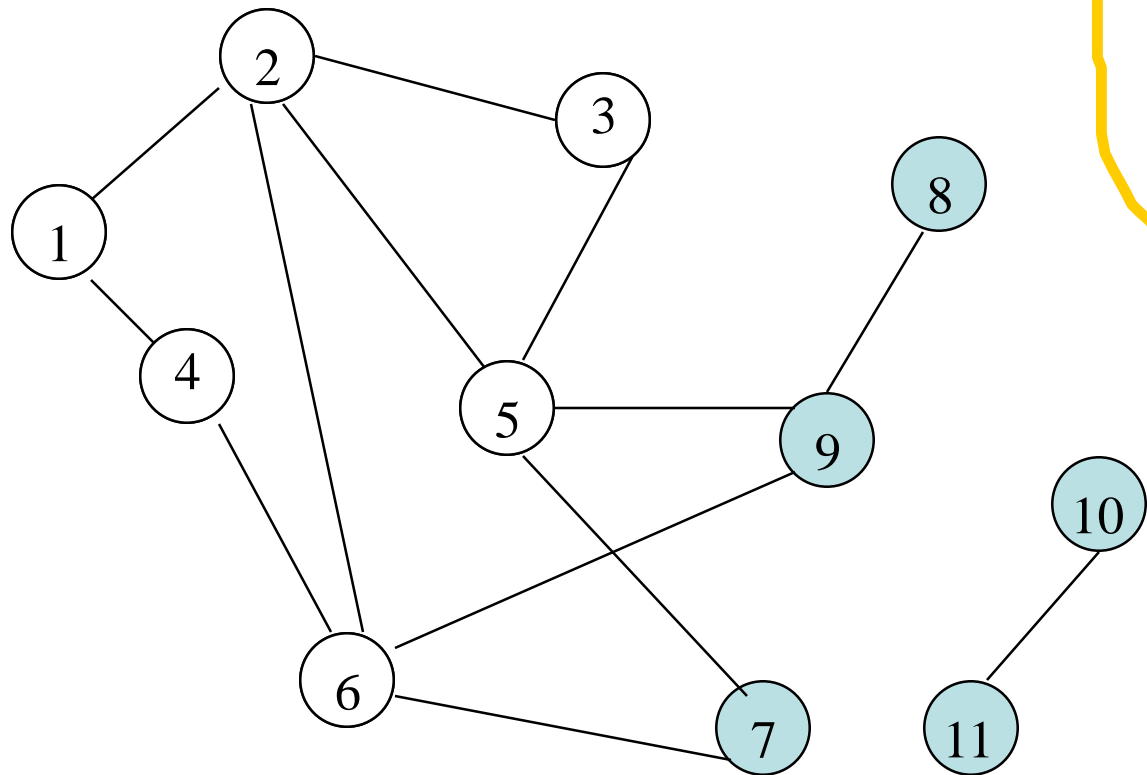
FIFO дараалал

2 4

Q -с 2 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



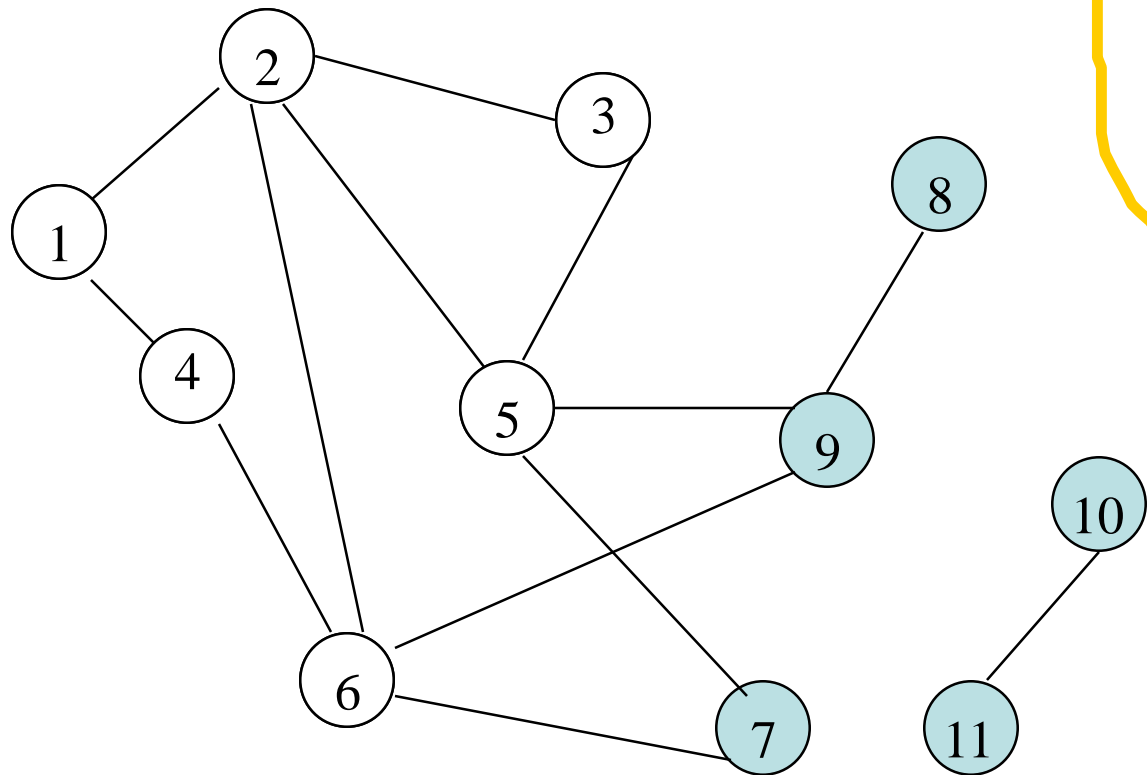
FIFO дараалал

4 5 3 6

Q -с 2 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



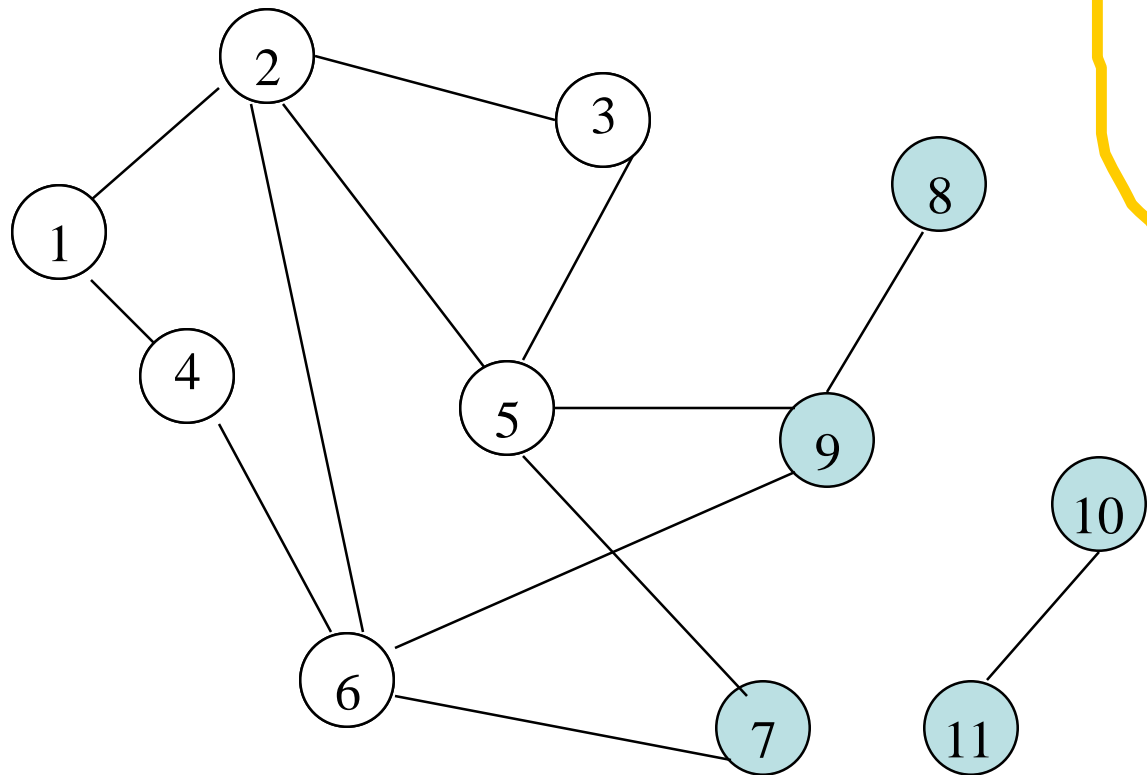
FIFO дараалал

4 5 3 6

Q -с 4 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



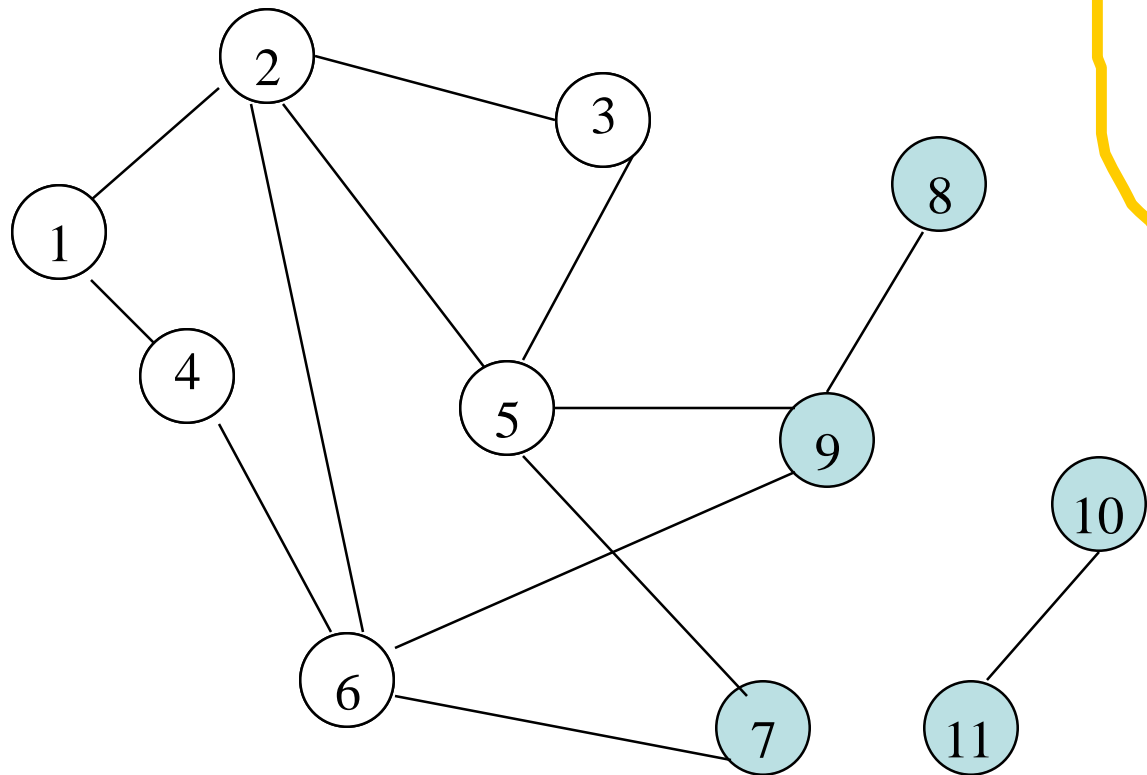
FIFO дараалал

5 3 6

Q -с 4 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



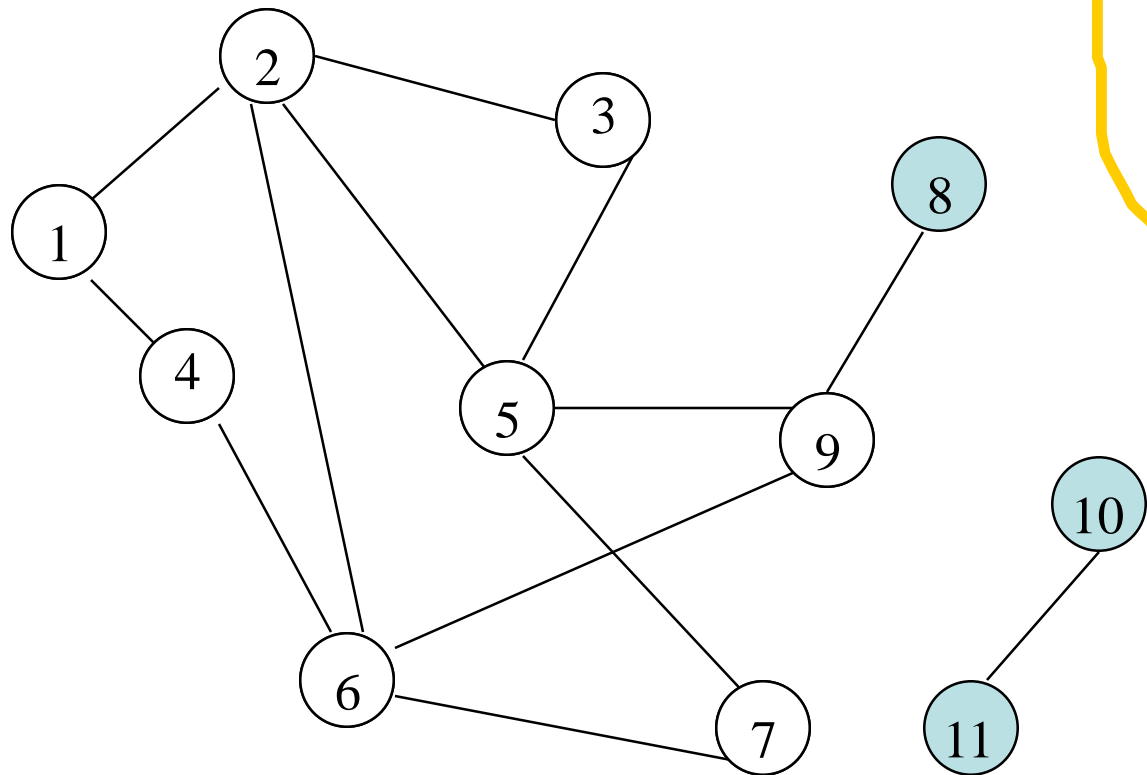
FIFO дараалал

5 3 6

Q -с 5 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



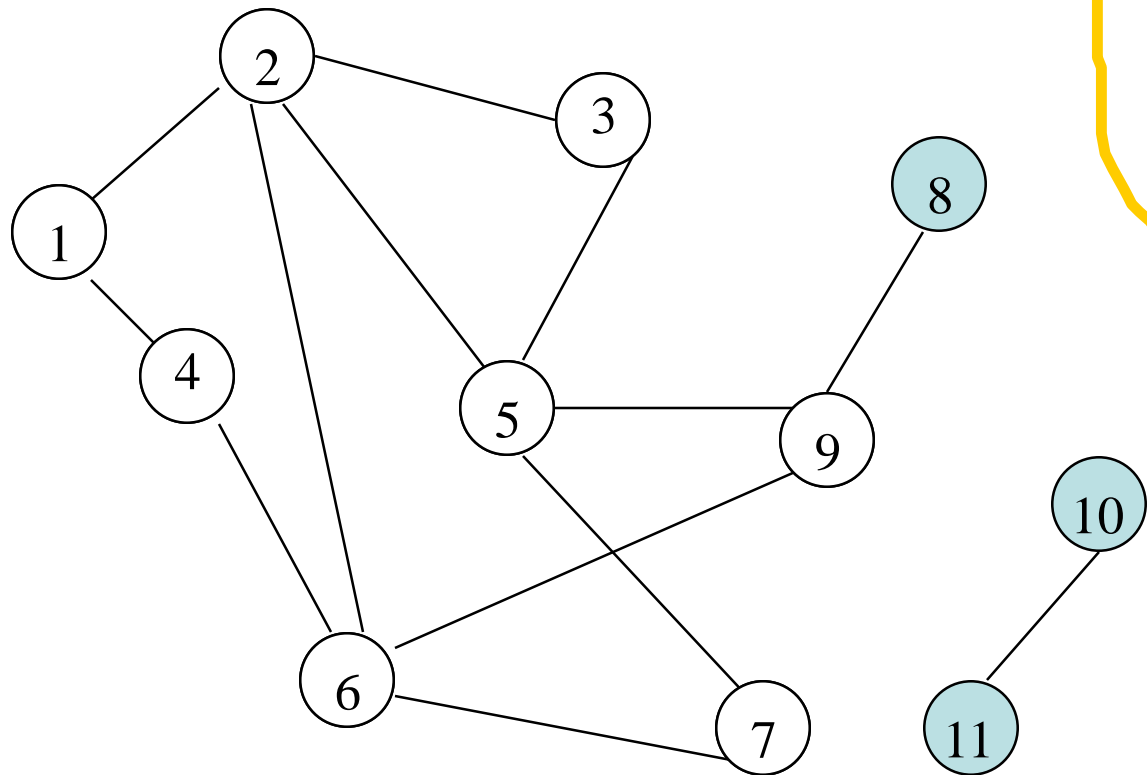
FIFO дараалал

3 6 9 7

Q -с 5 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



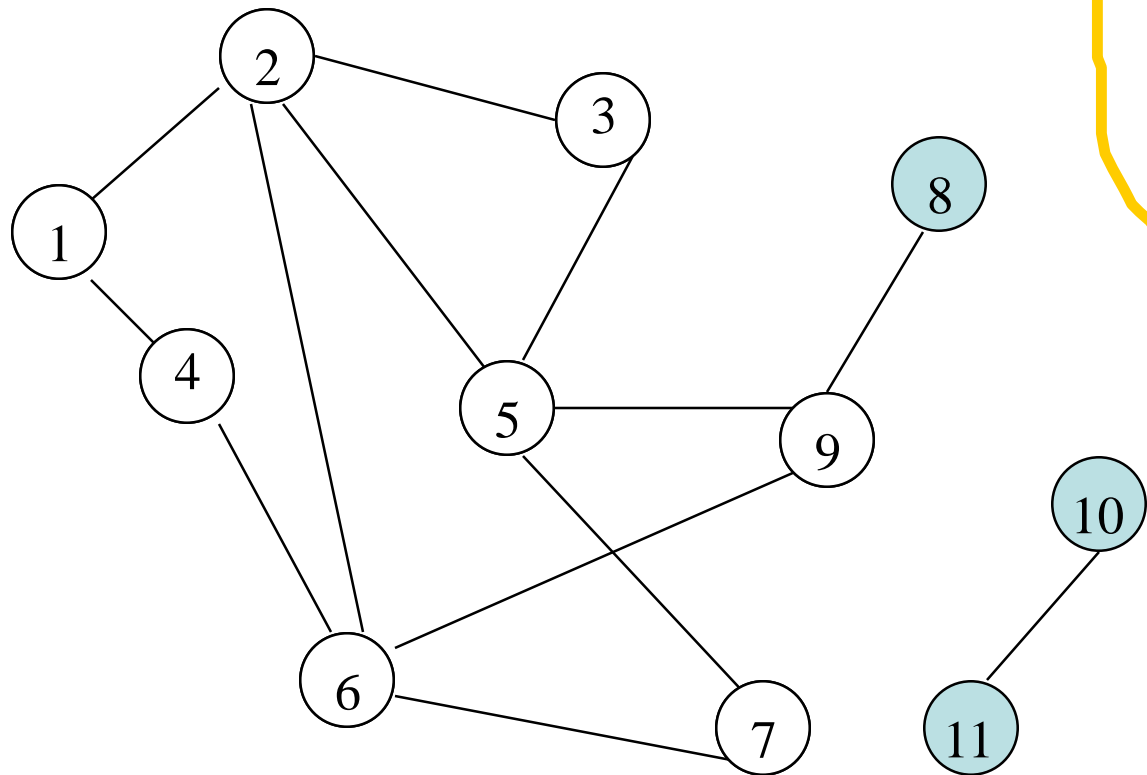
FIFO дараалал

3 6 9 7

Q -с 3 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



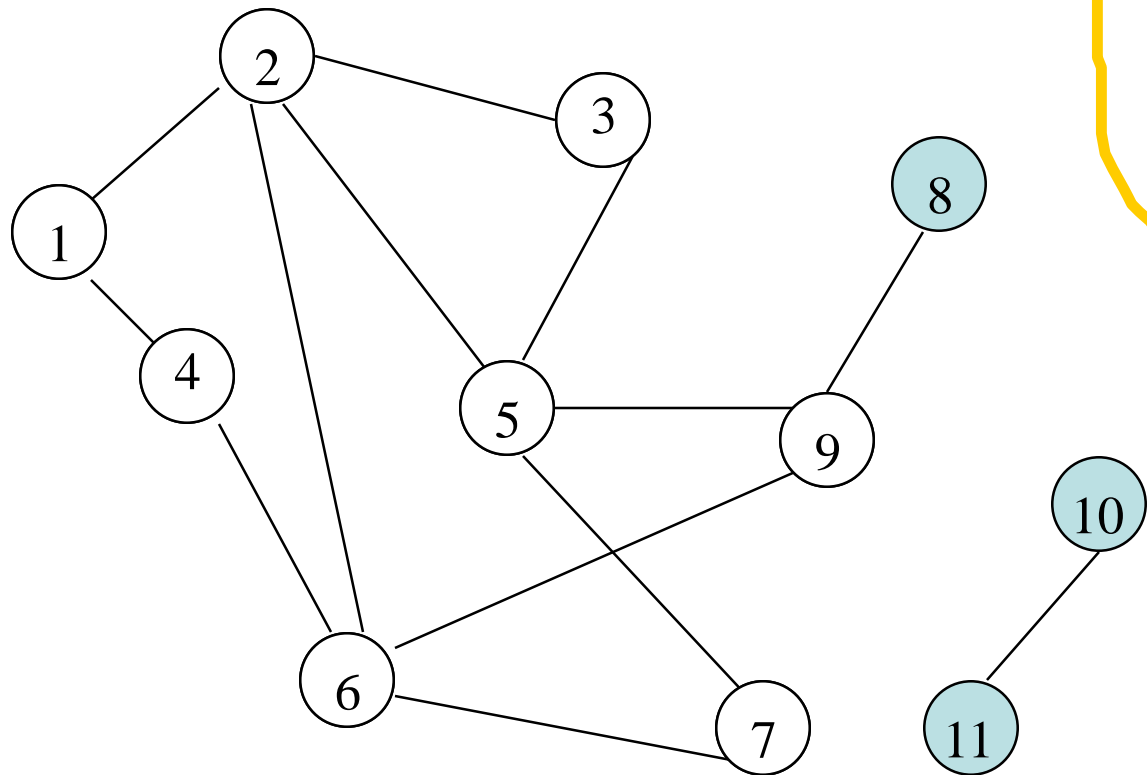
FIFO дараалал

6 9 7

Q -с 3 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



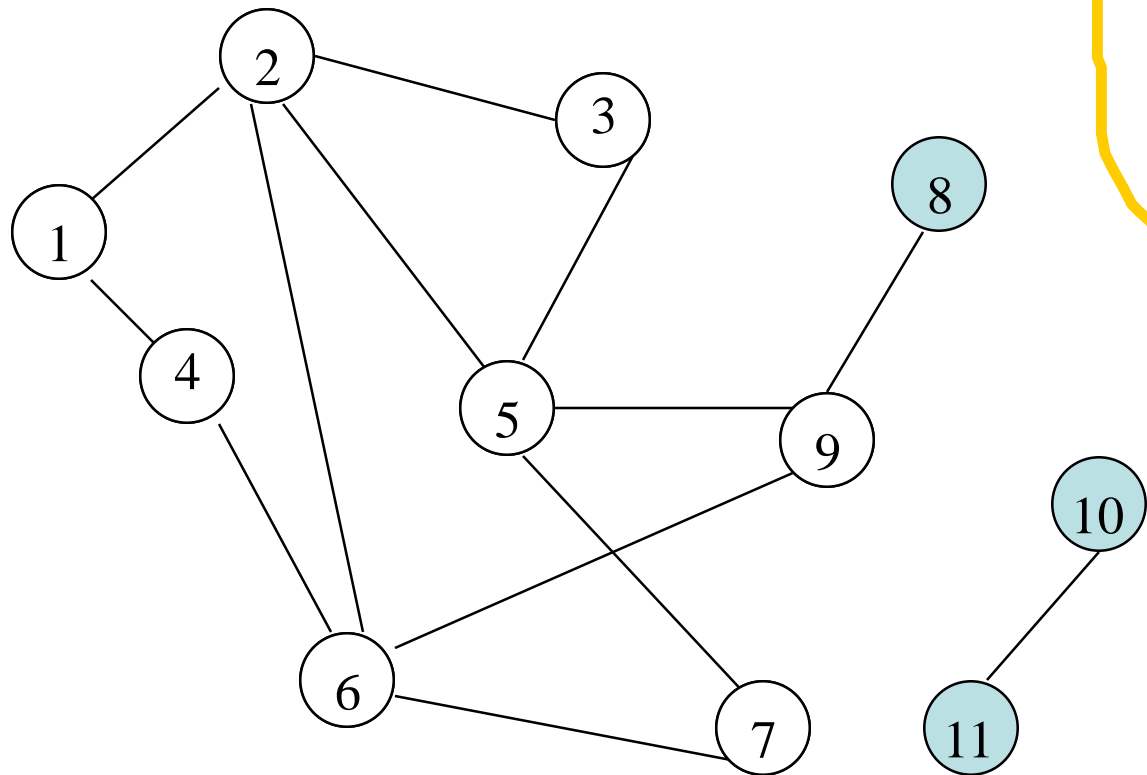
FIFO дараалал

6 9 7

Q -с 6 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



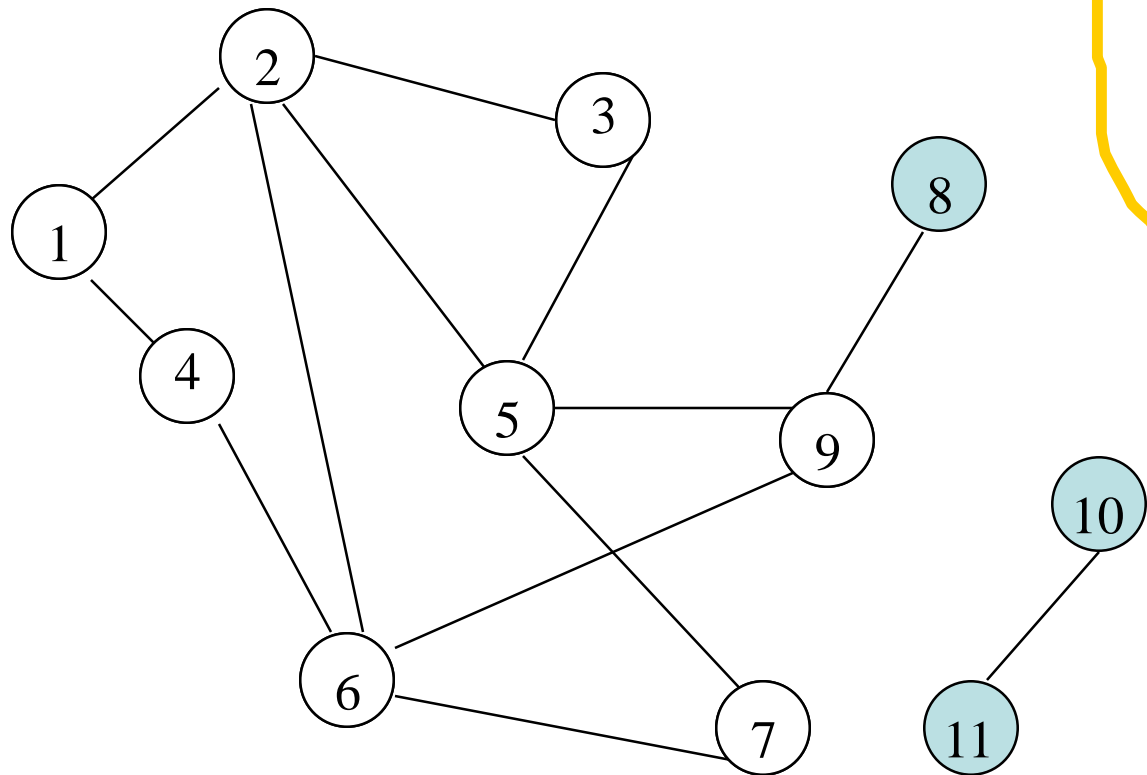
FIFO дараалал

9 7

Q -с 6 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



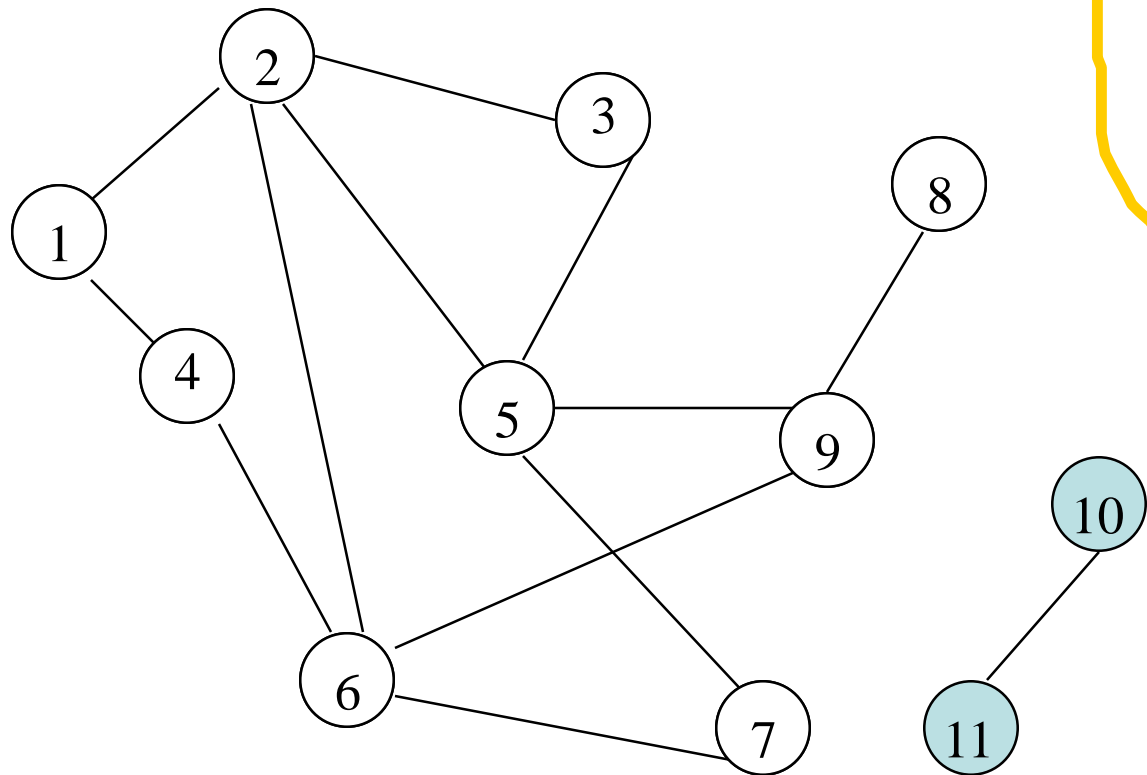
FIFO дараалал

9 7

Q -с 9 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



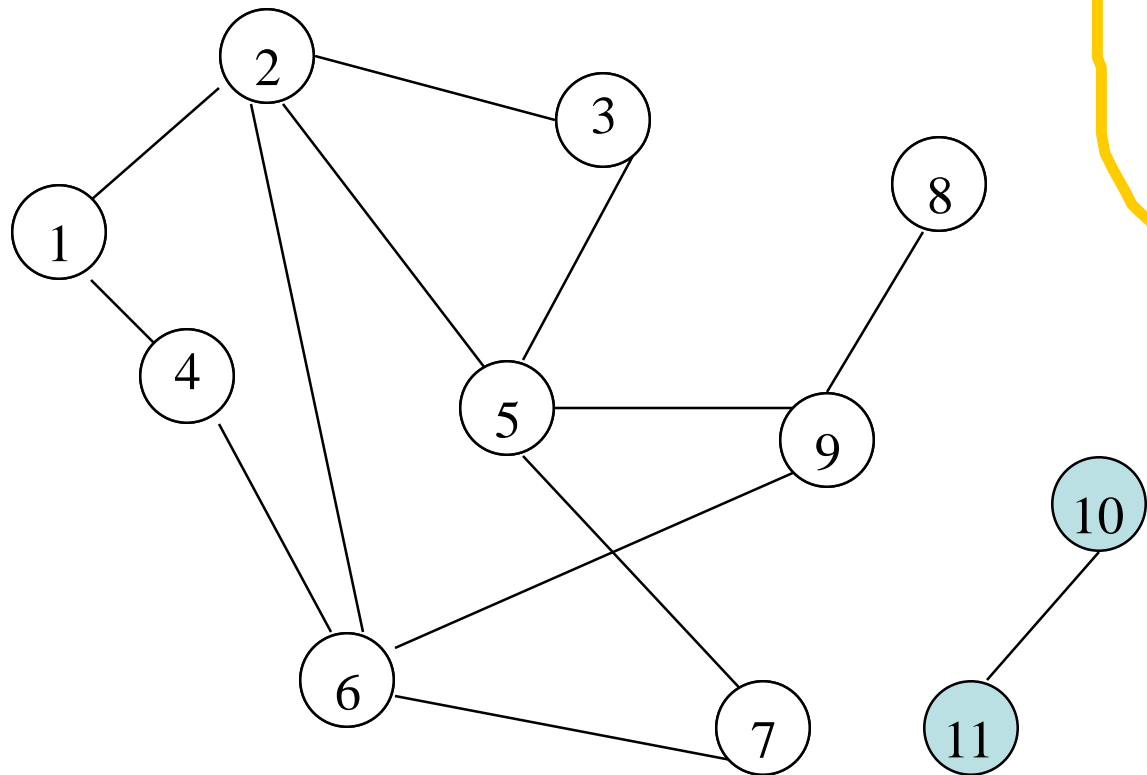
FIFO дараалал

7 8

Q -с 9 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



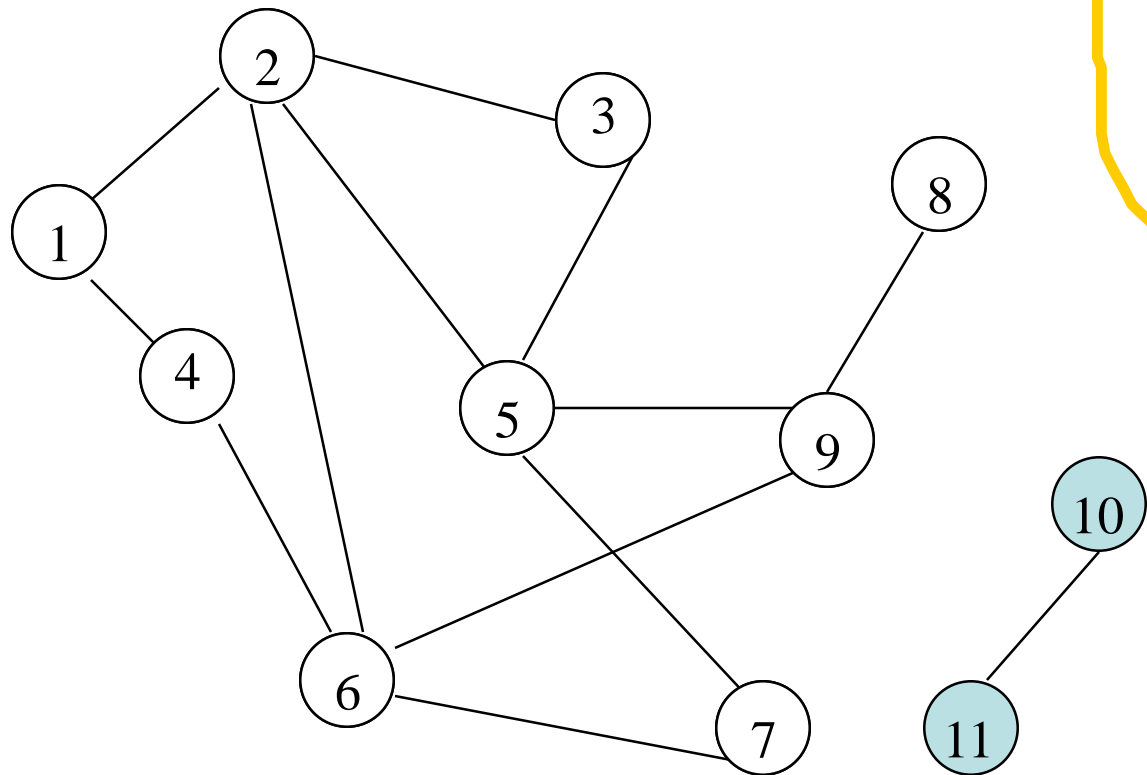
FIFO дараалал

7 8

Q -с 7 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



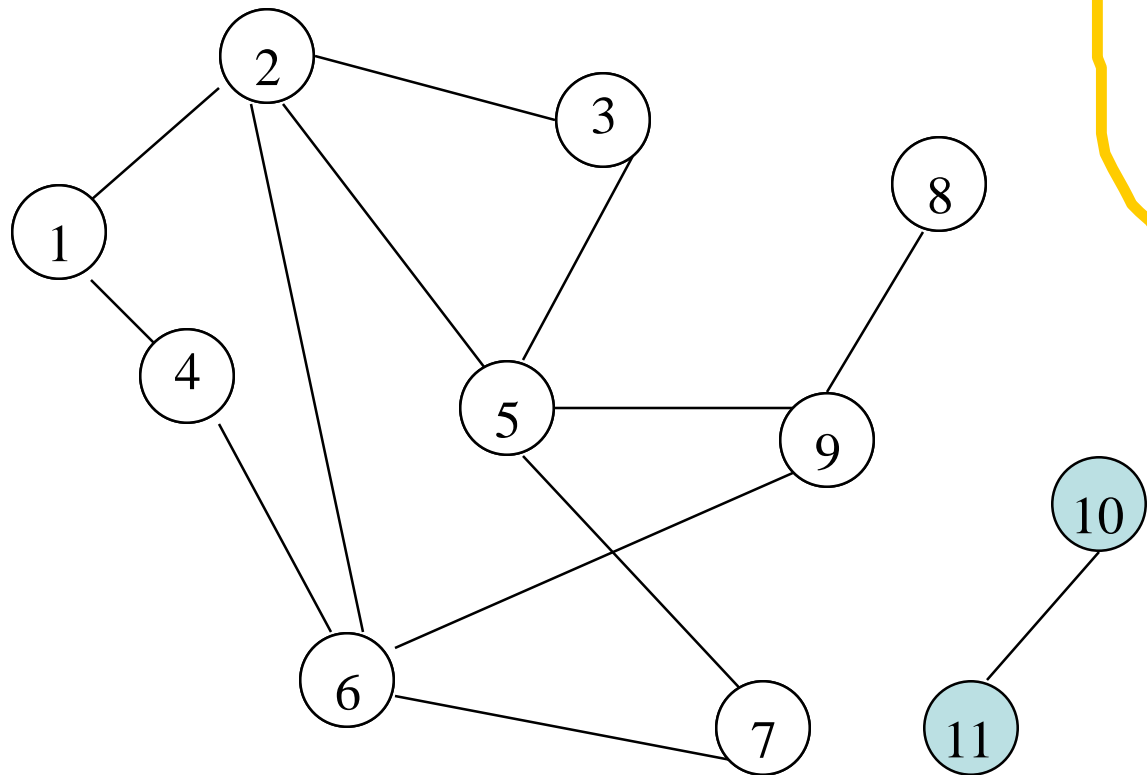
FIFO дараалал

8

Q -с 7 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



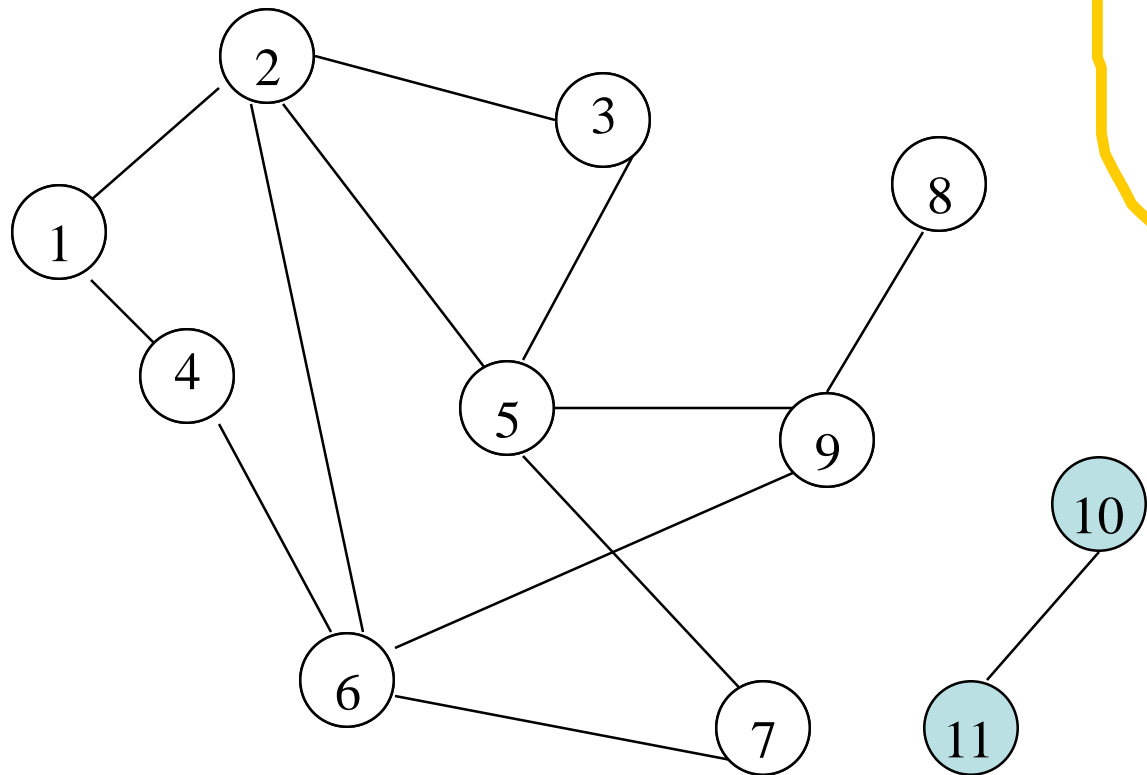
FIFO дараалал

8

Q -с 8 -г устгаж, айлчлаагүй хөршүүдээр нь айлчилж;

Q -д хийнэ.

Түвшний хайлтын жишээ



FIFO дараалал

Дараалал хоосон. Хайлт төгслөө.

Түвшний хайлтын шинж

- Эхний оройгоос хүрч болох бүх оройгоор (эхний оройг оролцуулаад) зочилдог.

Хугацаа



- Зочилсон орой бүрдараалалд яг нэг удаа орно (мөн тэндээс устгагдана) .
- Дарааллаас оройг устгахдаа, бид хөрш оройнуудыг нь шалгана.
 - $O(n)$ холболтын матриц ашиглахад
 - $O(\text{оройн зэрэг})$ холболтын жагсаалт ашиглахад
- Нийт хугацаа
 - $O(mn)$, үүнд m –хайлт хийж байгаа бүрдүүлбэр дэх оройн тоо (холболтын матриц)

Хугацаа



- $O(n + \text{бүрдүүлбэрийн оройн зэргийн нийлбэр})$ (холболтын жагсаалт)
= $O(n + \text{бүрдүүлбэрийн ирмэгийн тоо})$

Орой v –с орой u хүрэх зам

- Түвшний хайлтыг v оройгоос эхэлнэ
- Орой u –р зочилсон, эсхүл Q хоосорсон тохиолдолд зогсоно (Алин ч эхэлж тохиолдож болно).
- Хугацаа
 - $O(n^2)$ холболтын матриц ашиглахад
 - $O(n+e)$ холболтын жагсаалт ашиглахад (e –ирмэгийн тоо)

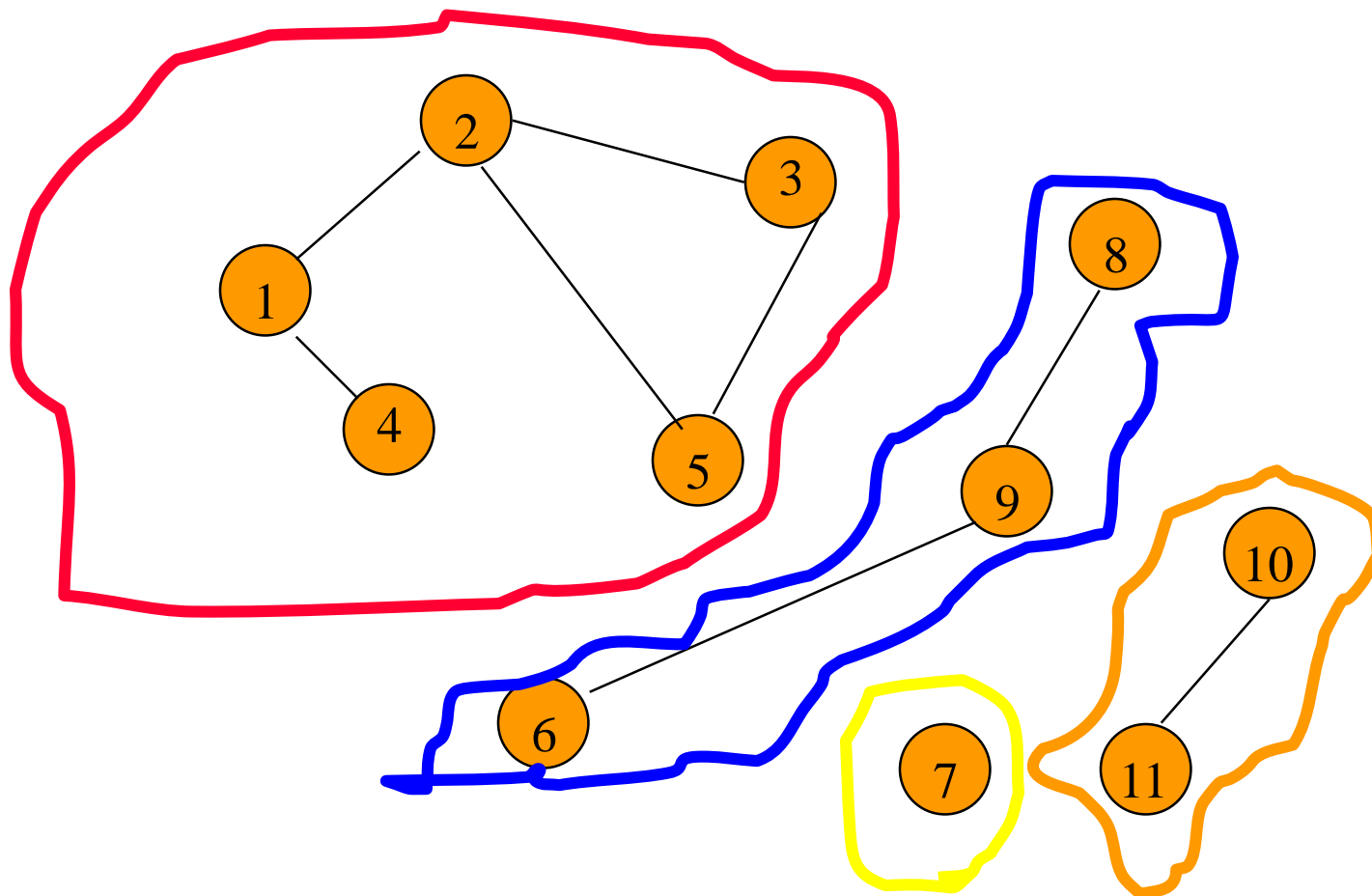
Граф холбогдсон уу?

- Графын дурын оройгоос Түвшний хайлтыг эхэлнэ.
- Бүх n оройгоор зочилсон бол граф холбогдсон.
- Хугацаа
 - $O(n^2)$ холболтын матриц ашиглахад
 - $O(n+e)$ холболтын жагсаалт ашиглахад (e – ирмэгийн тоо)

Холбогдсон бүрдүүлбэр

- Графын дурын зочлоогүй оройгоос Түвшнгий хайлт эхэлнэ.
- Шинээр айлчилсан оройнууд (тэдний хоорондох ирмэгүүдийг нэмээд) бүрдүүлбэрийг тодорхойлно.
- Бүх оройгоор зочилтол давтана.

Холбогдсон бүрдүүлбэрүүд

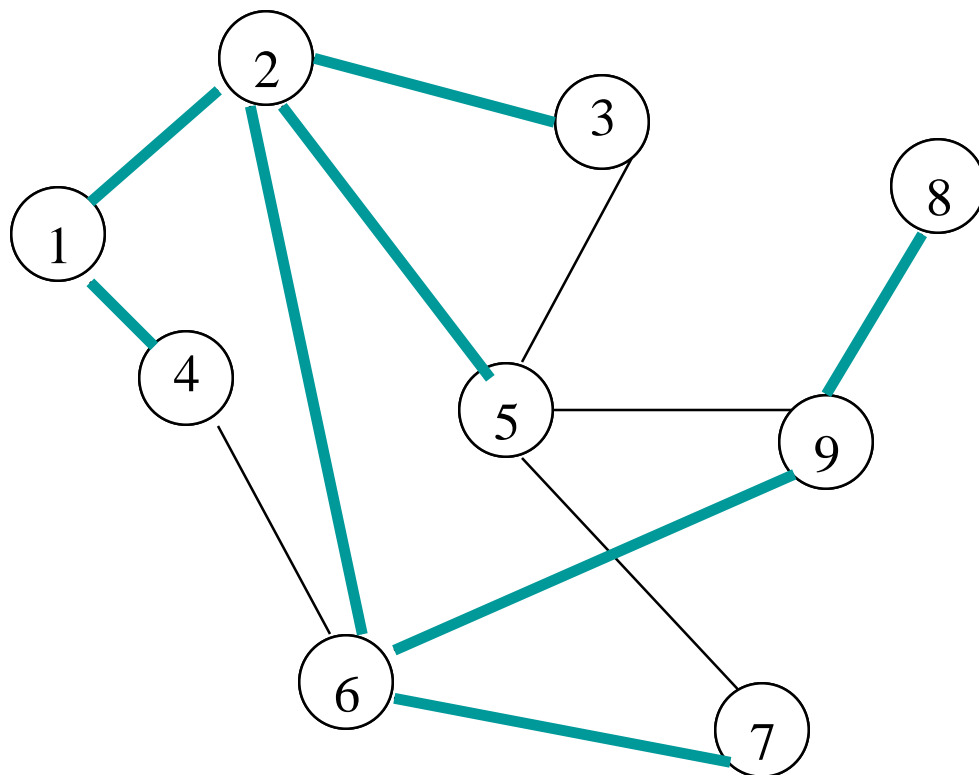


Хугацаа



- $O(n^2)$ холболтын матриц ашиглахад
- $O(n+e)$ холболтын жагсаалт ашиглахад (e – ирмэгийн тоо)

Бүрхэгч мод



1 –р оройгоос Түвшний хайлт хийнэ.

Түвшний бүрхэгч мод.

Бүрхэгч мод

- Графын дурын оройгоос Түвшний хайлт хийж эхэлнэ.
- Хэрвээ граф холбогдсон бол, $n-1$ ирмэгийг зочлоогүй оройнууд бүрхэгч модыг тодорхойлоход ашиглана (түвшний хайлтын бүрхэгч мод).
- Хугацаа
 - $O(n^2)$ холболтын матриц ашиглахад
 - $O(n+e)$ холболтын жагсаалт ашиглахад (e – ирмэгийн тоо)

Гүний хайлт

(Depth-First Search)

```
depthFirstSearch(v)
```

```
{
```

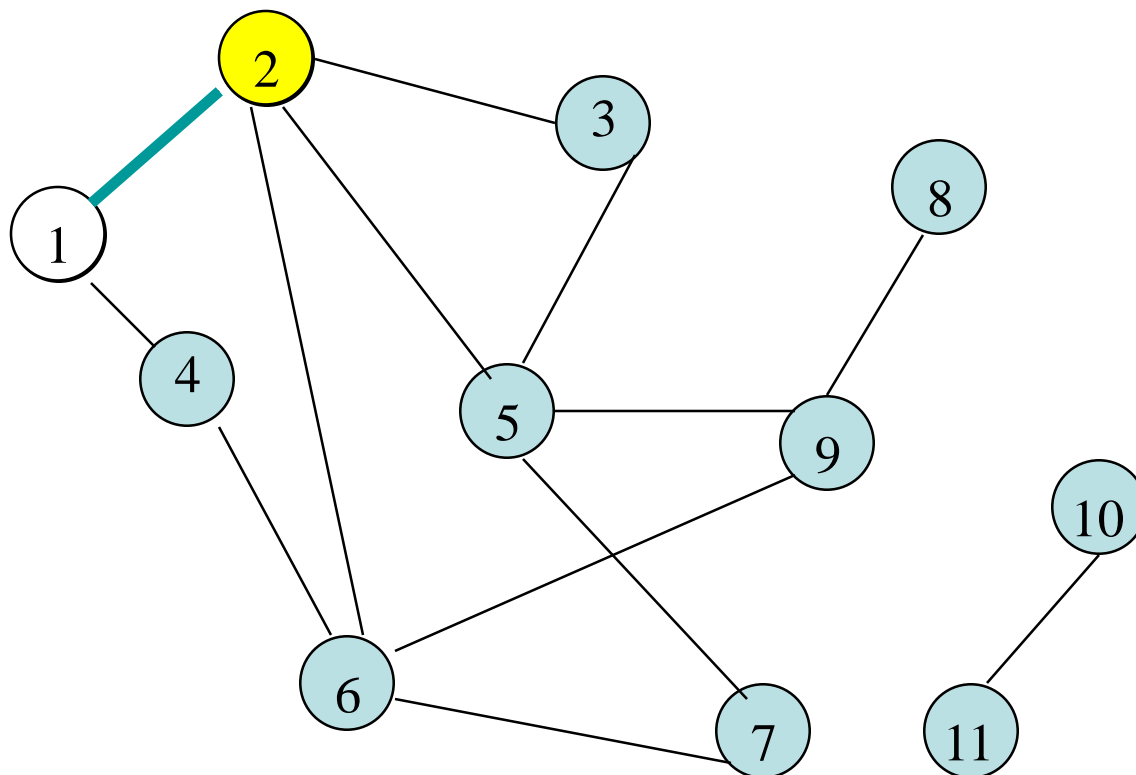
```
    Орой v –г зочилсон болгох.
```

```
    for (v -тэй хөрш зочлоогүй орой u  
        -н хувьд)
```

```
        depthFirstSearch(u);
```

```
}
```

Гүний хайлтын жишээ

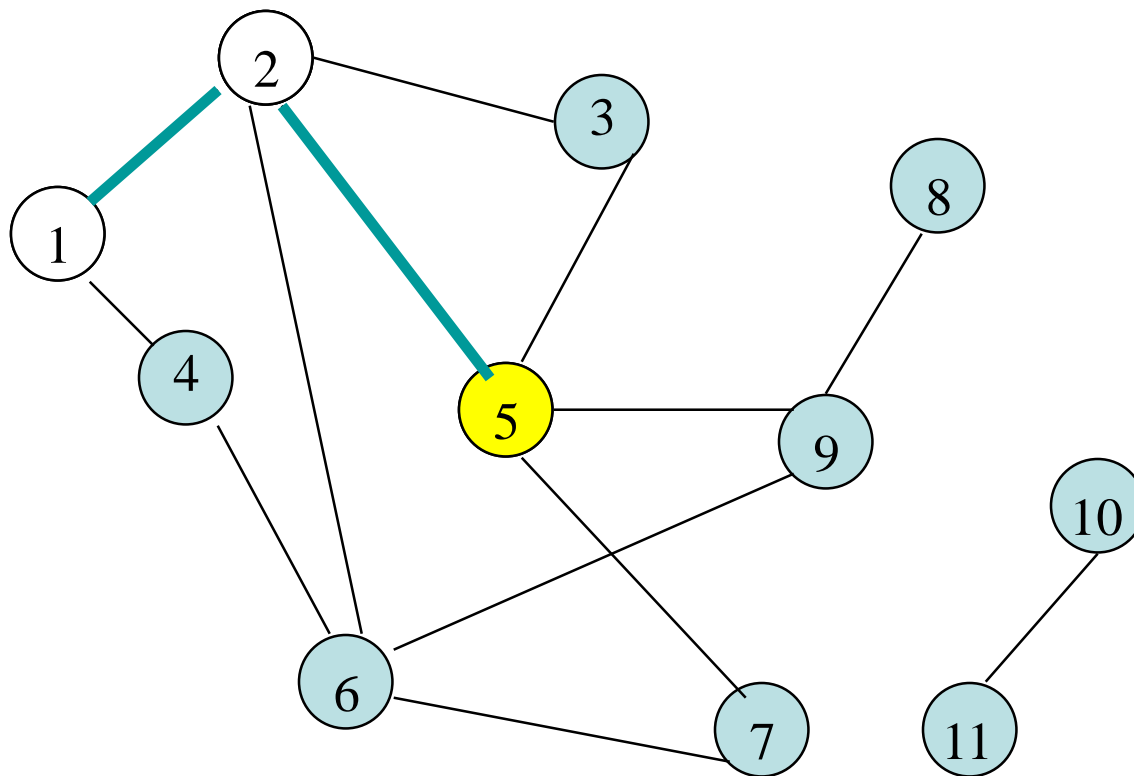


Хайж эхлэх орой - 1.

Орой 1 –г тэмдэглээд 2 эсхүл 4 –с гүний хайлт эхлүүлнэ.

Орой 2 –г сонгосон гэж үзье.

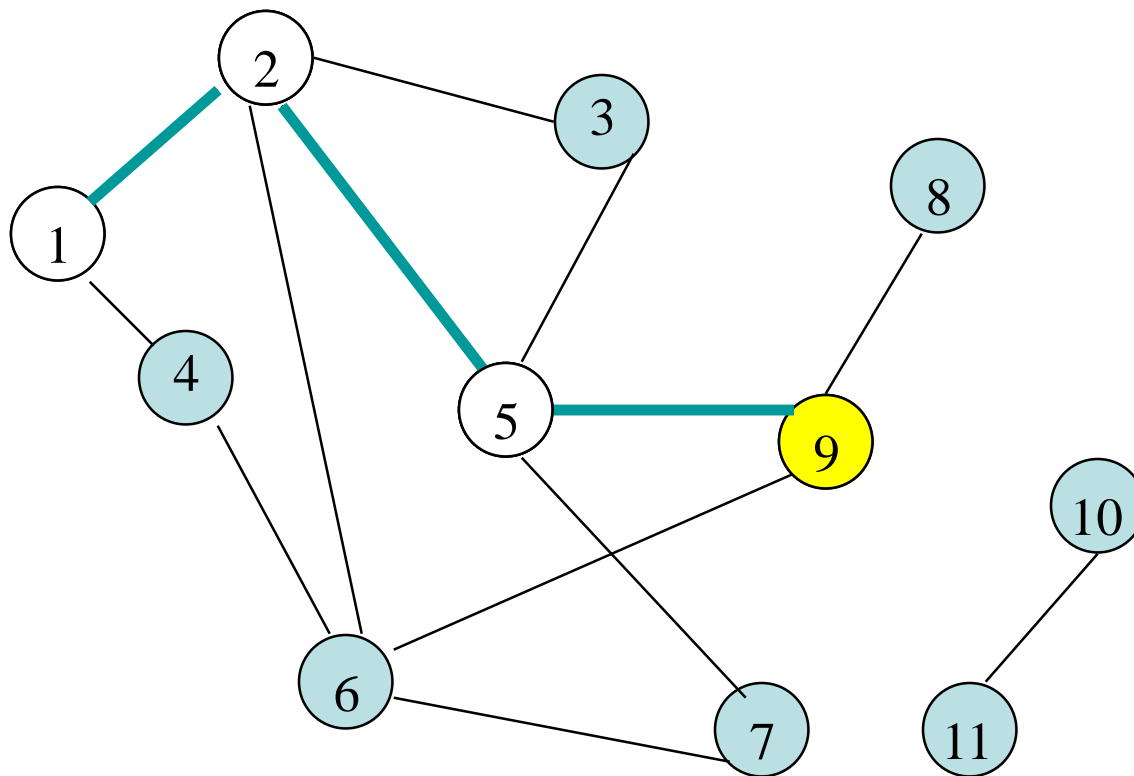
Гүний хайлтын жишээ



Орой 2 –г тэмдэглээд 3, 5, эсхүл 6 -с гүний хайлт эхлүүлнэ

Орой 5 –г сонгосон гэж үзье.

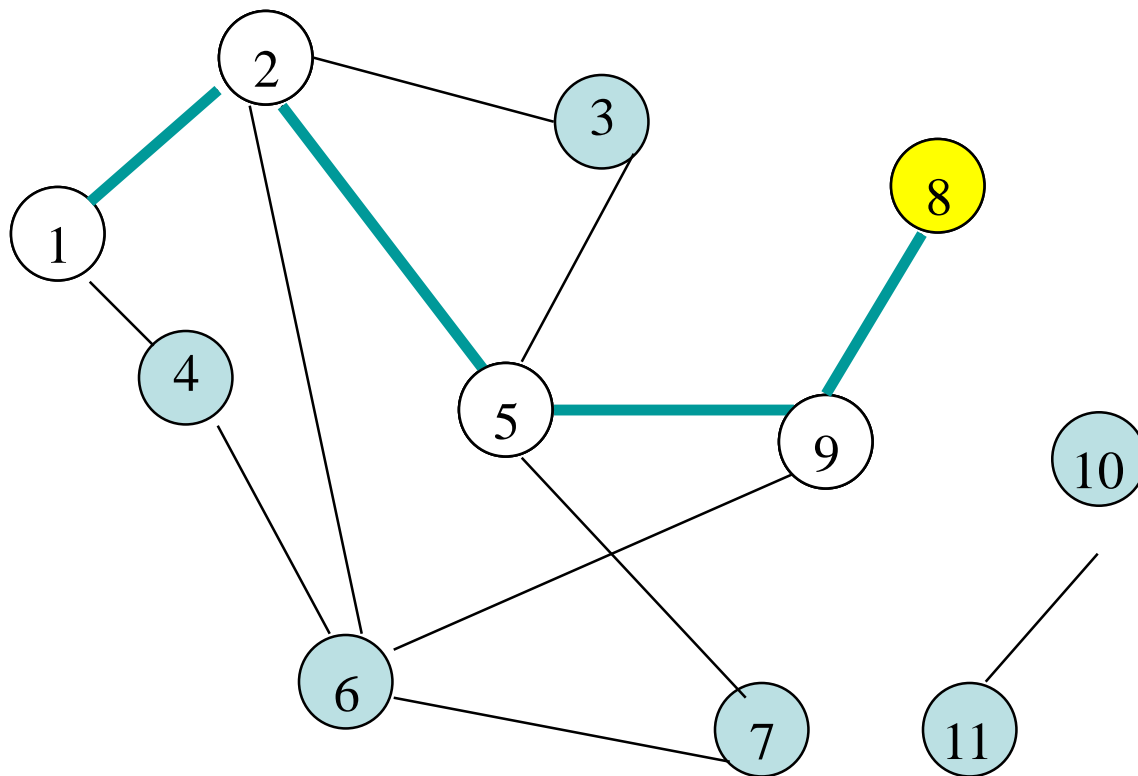
Гүний хайлтын жишээ



Орой 5 –г тэмдэглэж 3, 7, эсхүл 9 -с гүний хайлт эхлүүлнэ

Орой 9 –г сонгосон гэж үзье.

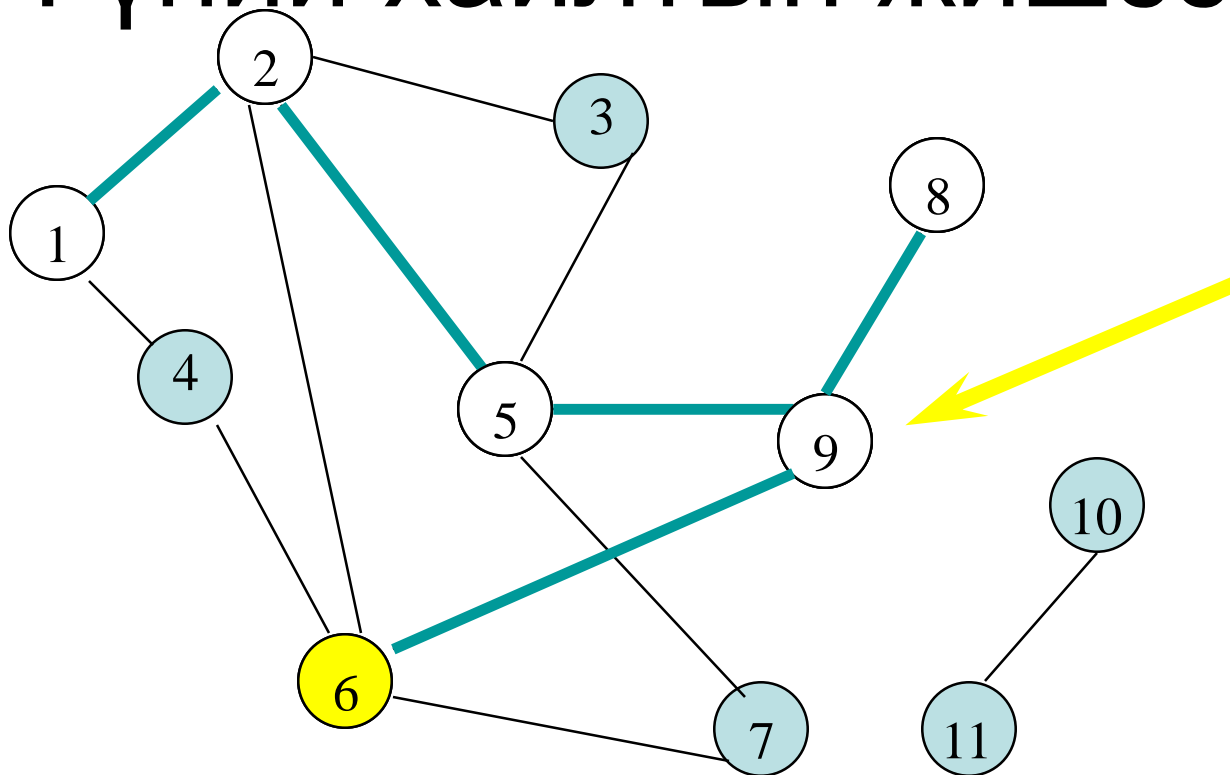
Гүний хайлтын жишээ



Орой 9 –г тэмдэглээд 6 эсхүл 8 -с гүний хайлт эхлүүлнэ

Орой 8 –г сонгосон гэж үзье.

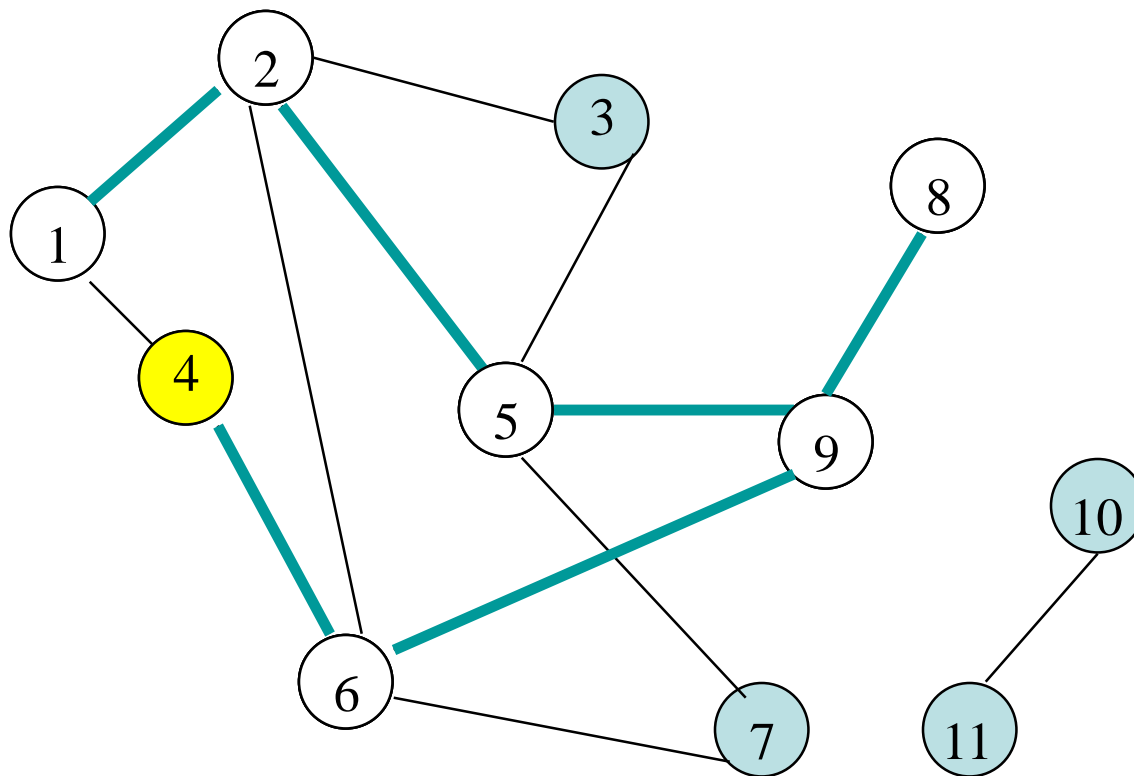
Гүний хайлтын жишээ



Орой 8 –г тэмдэглээд орой 9 рүү буцна.

Орой 9 -с `dfs(6)`.

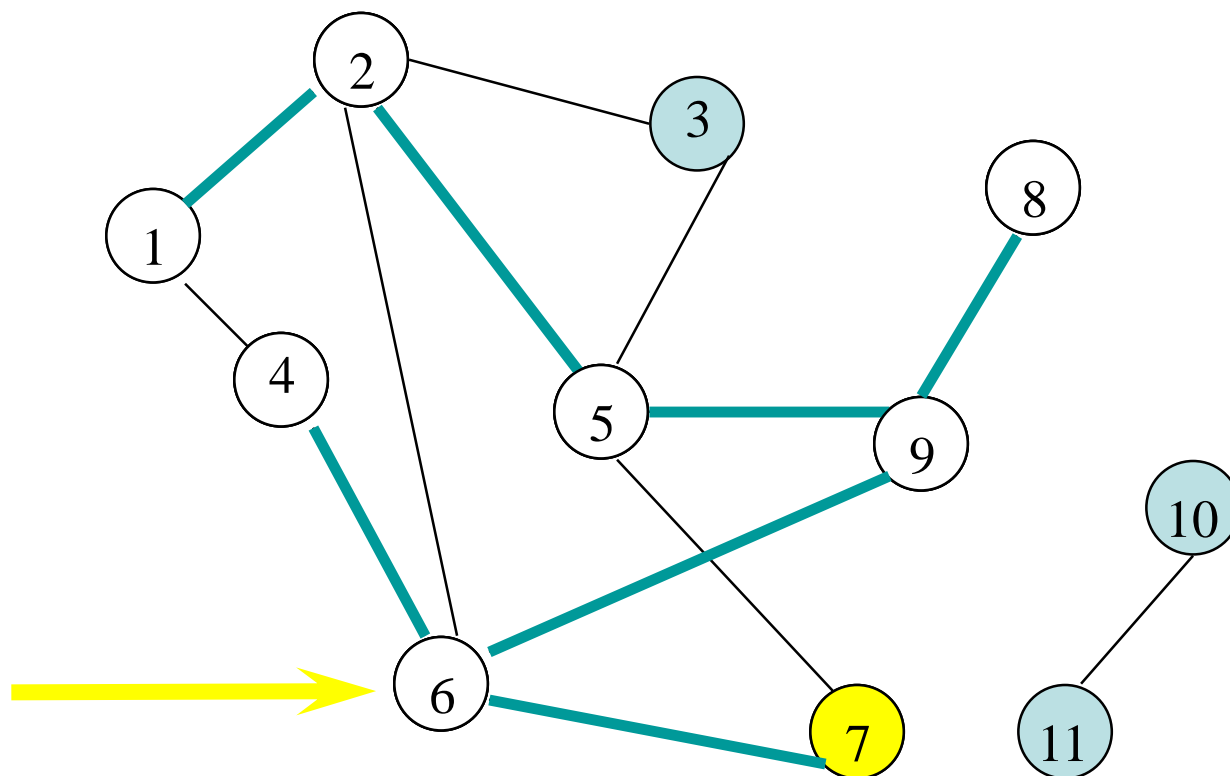
Гүний хайлтын жишээ



Орой 6 –г тэмдэглээд 4 эсхүл 7 -с гүний хайлт
эхлүүлнэ

Орой 4 –г сонгосон гэж үзье.

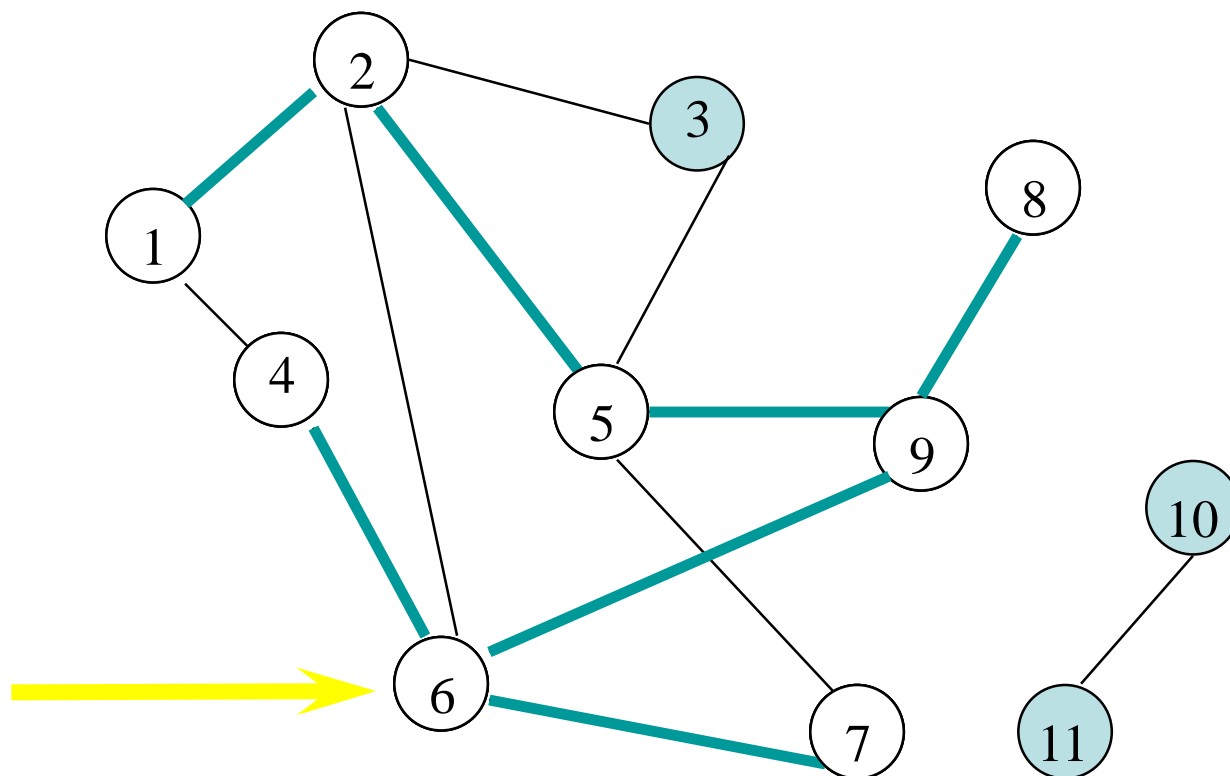
Гүний хайлтын жишээ



Орой 4 –г тэмтэглээд орой 6 -рүү буцна

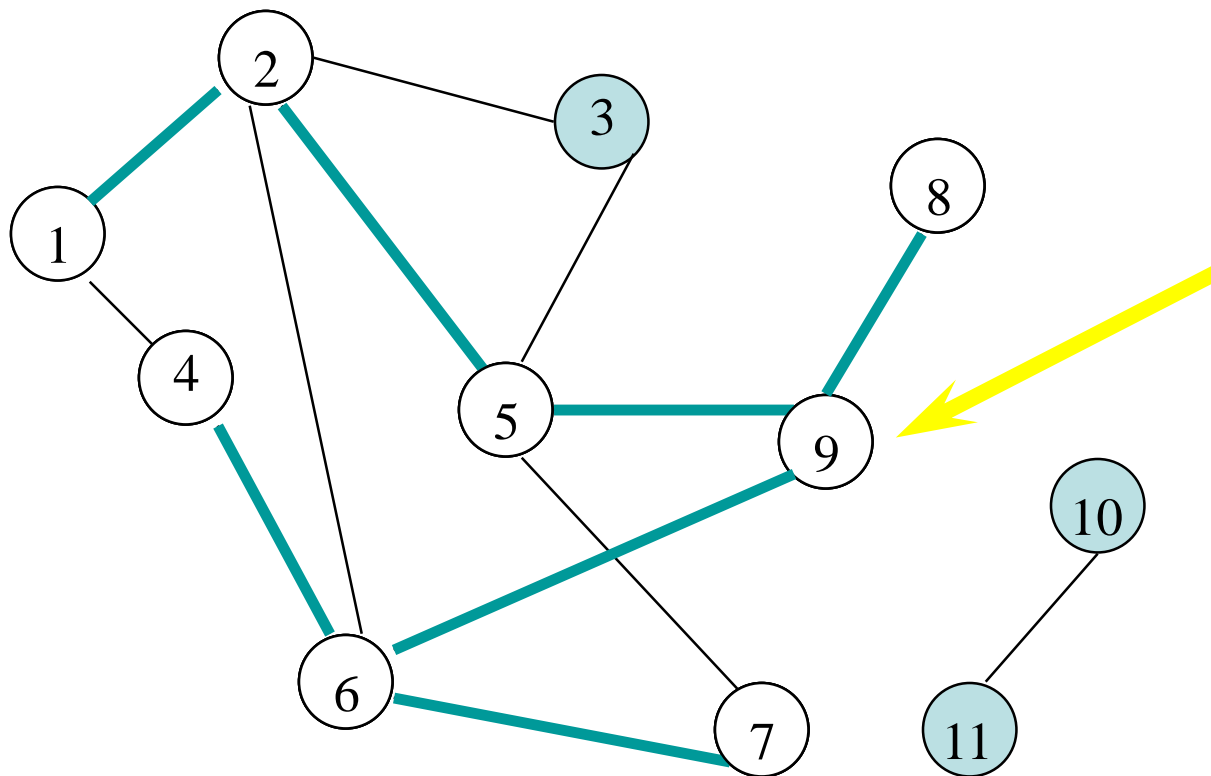
Орой 6 -с $\text{dfs}(7)$.

Гүний хайлтын жишээ



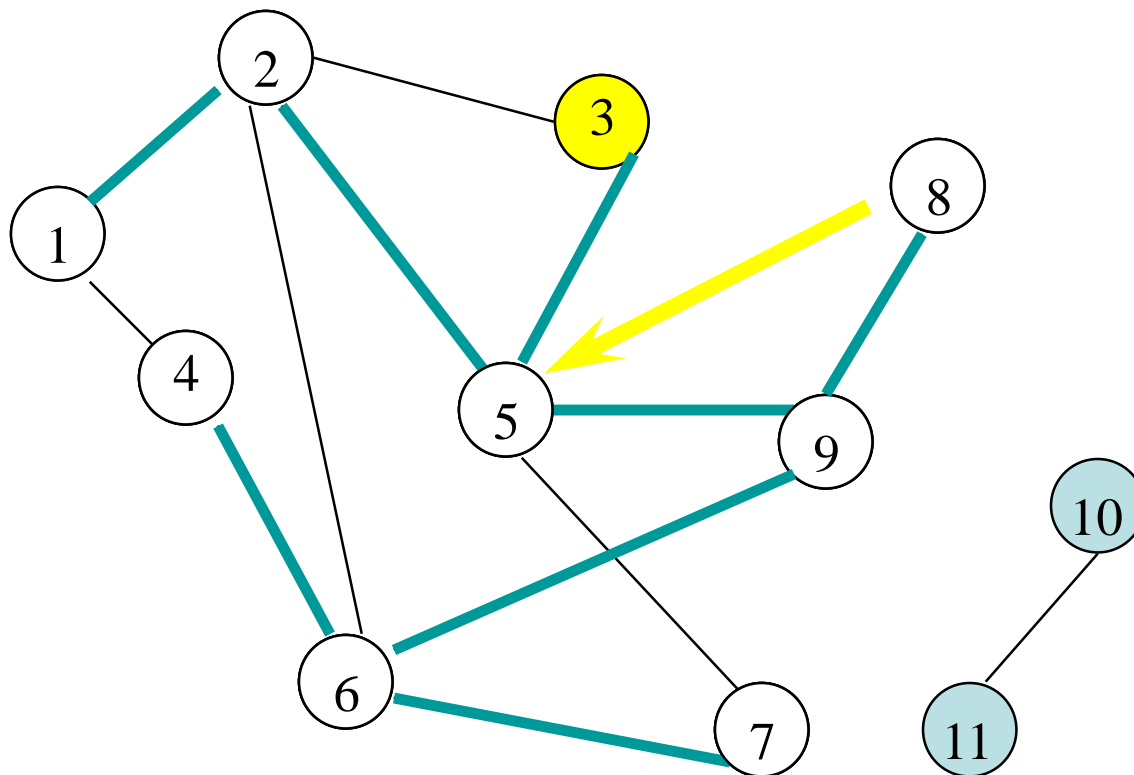
Орой **7** –г тэмдэглээд орой **6** -рүү буцна
9 рүү буцна

Гүний хайлтын жишээ



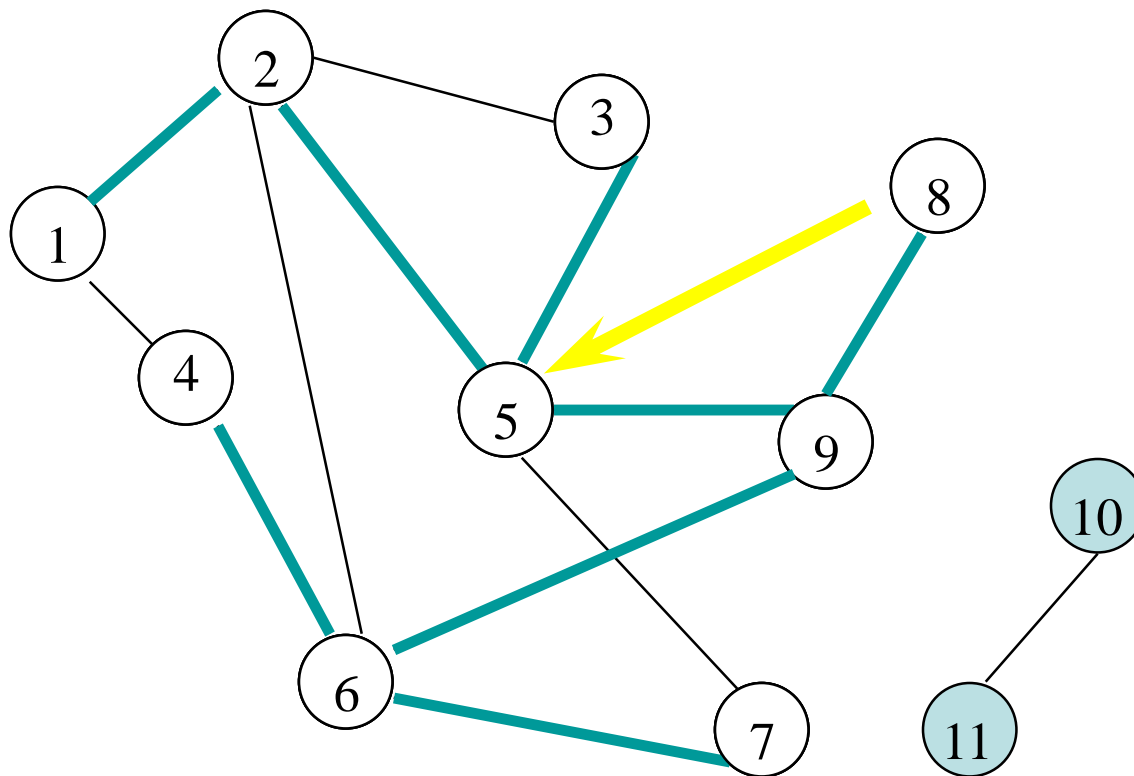
5 -рүү буцна

Гүний хайлтын жишээ



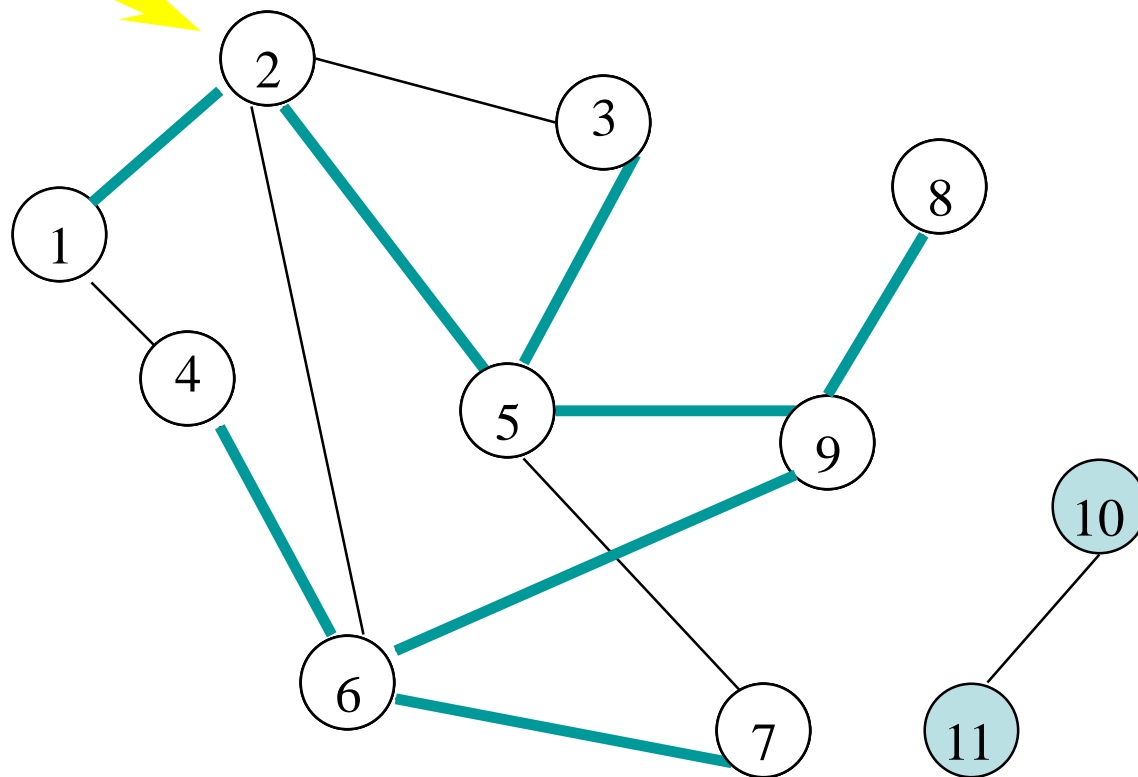
dfs(3).

Гүний хайлтын жишээ



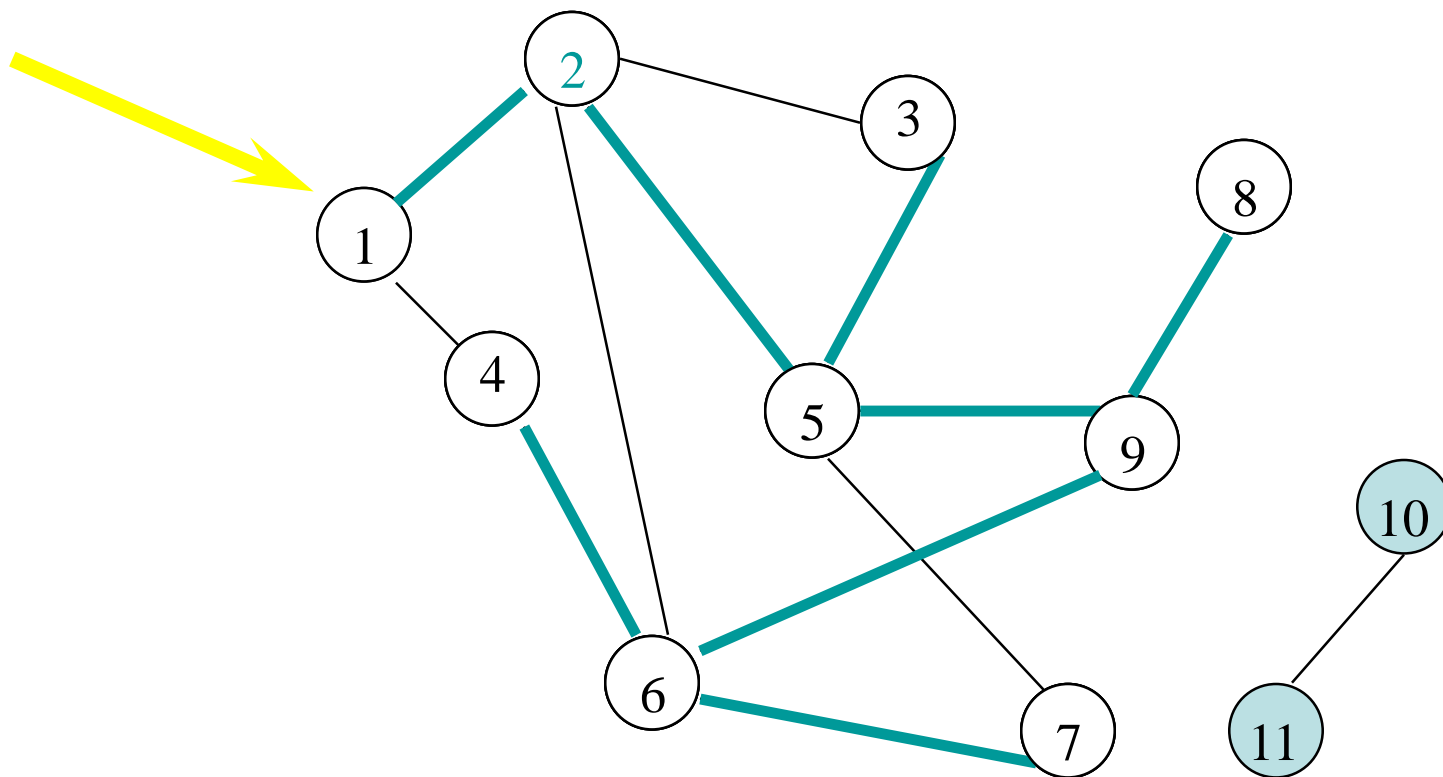
Орой 3 –г тэмдэглээд орой 5 -рүү буцна
2 -рүү буцна

Гүний хайлтын жишээ



1 -рүү буцна

Гүний хайлтын жишээ



Дуудсан функц руу буцна.

Гүний хайлтын шинж

- Хугацааны хувьд BFS -тай адил.
- Зам хайх, холбогдсон бүрдүүлбэр, бүрхэгч модны хувьд адилхан шинжтэй.
- Граф холбогдсон бол тэмдэглээгүй оройнуудад хүрэх ирмэгүүд гүний хайлтын бүрхэгч модыг тодорхойлно.
- Зарим бодлогод bfs, заримд нь dfs илүү тохирдог.