TOSHKENT AMALIY FANLAR UNIVERSITETI

Ma'lumotlar tuzilmasi va algoritmlar fani

"Kompyuter injiniring" kafedrasi

Katta o'qituvchi Kendjayeva Dildora Xudayberganovna

Grafda o‘tish algoritmlari Grafda o‘tish eni bo‘yicha qidiruv- BFS algoritmi, Grafda o‘tish bo‘yi bo‘yicha qidiruv (DFS) algoritmi

7 - Ma’ruza

MA’RUZA REJASI

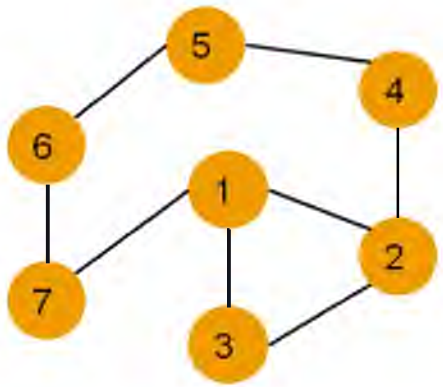
Grafda o‘tish eni bo‘yicha qidiruv - BFS algoritmi

Grafda o‘tish bo‘yi bo‘yicha qidiruv (DFS) algoritmi

Misollar

Graflar bilan ishlash

- Graflar bilan ishlashda barcha asosiy amallar (masalan, grafni bitta ko‘rinishda ikkinchisiga o‘tkazish, bosib chiqarish yoki grafni chizish) uning tizimli o‘tishini, ya’ni grafning har bir uchiga va har bir qirrasiga tashrif buyurishni nazarda tutadi.
- Grafni bosib o‘tish jarayonida har bir uch uchta holatdan birida bo‘ladi:
  1) ochilmagan - uchning dastlabki holati;
  2) ochiq - uch topilgan, ammo unga tushgan qirralar ko‘rib chiqilmagan;
  3) ishlov berilgan (belgilangan) - ushbu uchga tushgan barcha qirralarga tashrif buyuriladi.
- Grafning har bir uchi ketma-ket ushbu holatlarning barchasini qabul qilishi aniq. Dastlab, faqat bitta uch ochiq bo‘ladi, ya’ni grafni bosib o‘tish ushbu uchdan boshlanadi.

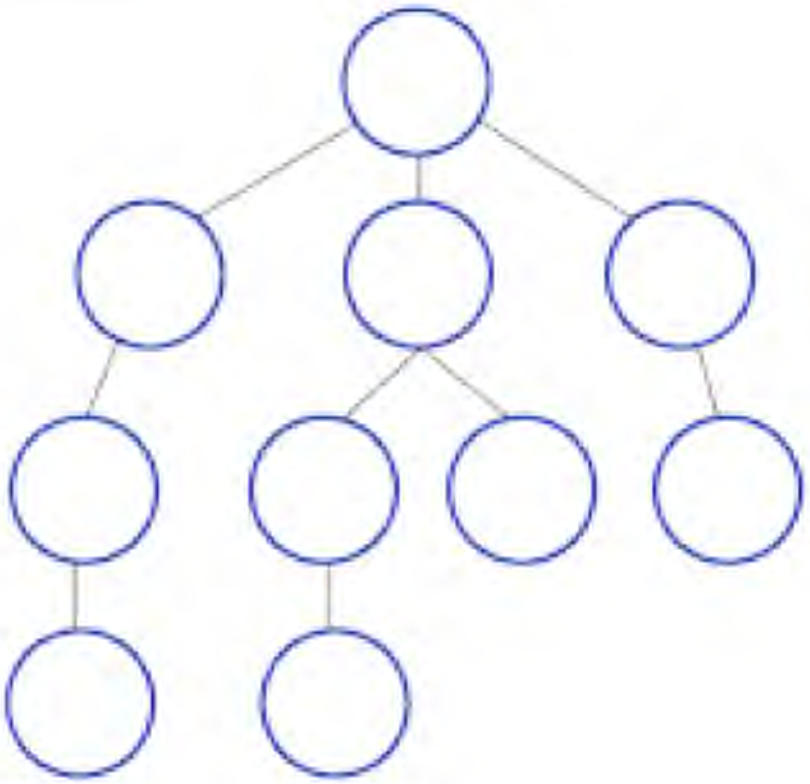


Grafda o‘tish eni bo‘yicha qidiruv - BFS algoritmi.

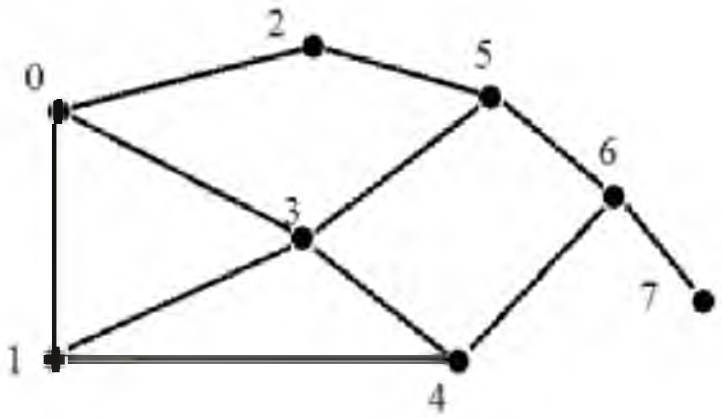
- G = (V, E) grafi berilgan bo‘lsin va boshlang‘ich uchi v tanlansin. Birinchi kenglik bo‘yicha qidirish algoritmi v uchga yetib boruvchi barcha uchlarni "ochish" uchun G grafning barcha qirralarini muntazam ravishda kesib o‘tadi.
- O‘tish jarayonida barcha uchlarni o‘z ichiga olgan dastlabki uchda joylashgan kenglik bo‘yicha qidiruv daraxtini yaratadi.

Grafda o‘tish eni bo‘yicha qidiruv - BFS algoritmi.

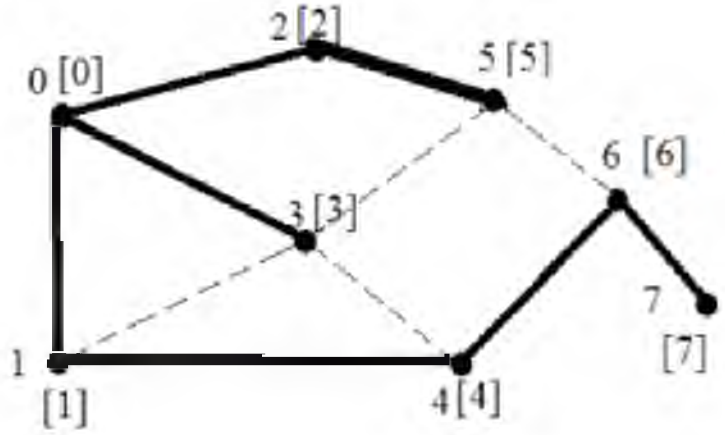
- E‘ tibor bering, ildiz uchidan ushbu daraxtning istalgan uchiga masofa (qirralarning soni) eng qisqa bo‘ladi.
- Kenglik bo‘yicha birinchi qidiruv ushbu nomga ega, chunki grafni bosib o‘tish jarayonida k+1 masofadagi har qanday uchni qayta ishlashdan oldin k masofadagi barcha uchlar belgilanadi.



- Ro‘yxatning birinchi yuqori qismi keyingi uchga aylanadi. Amaldagi bilan qo‘shni bo‘lgan avval belgilanmagan barcha uchning ro‘yxatning oxiriga qo‘shiladi (ochiladi).
- Joriy uch ro‘yxatdan o‘chirilib, 2 raqami bilan belgilanadi. Jarayon uchlar ro‘yxati bo‘sh bo‘lguncha davom etadi. Ma’lumotlar ro‘yxatining ushbu ko‘rinishi navbat deyiladi.



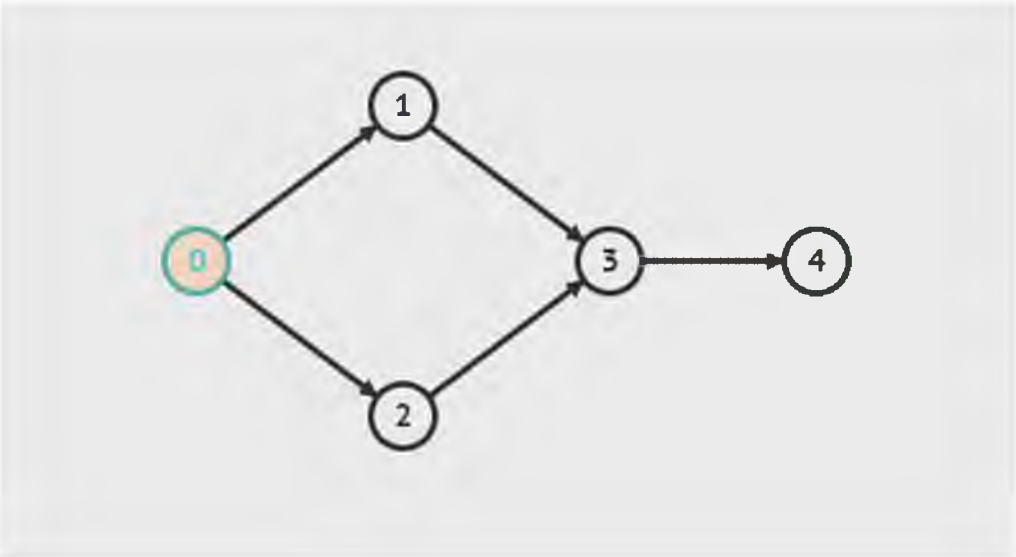
a)



b)

Grafda o‘tish eni bo‘yicha qidiruv - BFS algoritmi.

- Tasvirdan ko‘rinib turibdiki, algoritmning o‘zi juda ahamiyatsiz. Tashrif uchun uchlar navbati saqlanib qoladi.
- Keyingi uchga tashrif buyurganida, hali tashrif buyurmagan va hali navbatda bo‘lmagan barcha qo‘shnilari navbatga qo‘shiladi.
- Uchga allaqachon tashrif buyurilganligini tekshirish uchun bir qator yorliqlardan foydalaniladi.



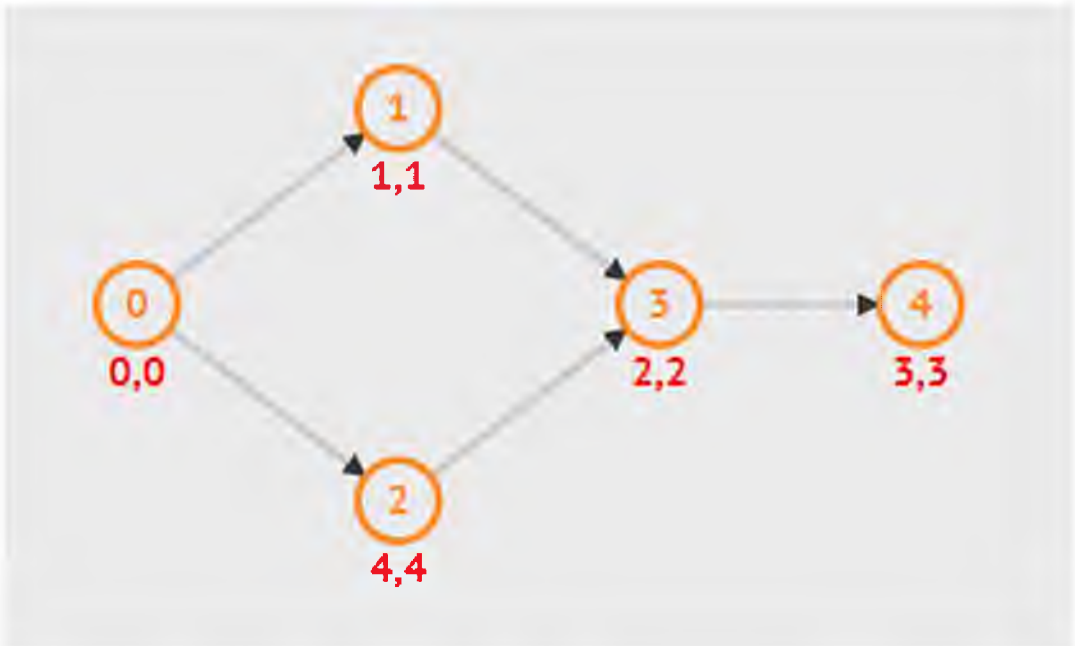
Dastlab, boshlang‘ich uchdan tashqari barcha **i** uchun **visited[i] = false** qiymatini qabul qiladi. **i** uch **visited[i]** navbatiga qo‘shilganda, **true** qiymati tayinlanadi.

```
#include <iostream>
using namespace std;

vector<int> graph[100000];
bool used[100000];

int main() {
  //Grafni kiritish
  // ... Bu qismda graf matritsa ko‘rinishida kiritiladi
  queue<int> q;
  q.push(0);
  used[0] = true;

  while (!q.empty()) {
      int cur = q.front();
      q.pop();

      cout << "BFS : " << cur + 1 << endl;

      for (int k: graph[cur]) {
          if (!used[k]) {
              q.push(k);
              used[k] = true;
          }
      }
  }
}
```

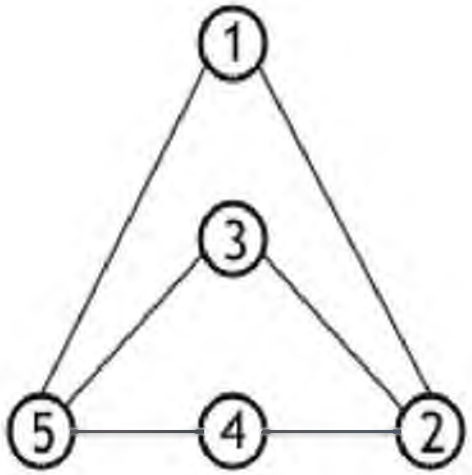
Ushbu algoritmning murakkabligi **O(n^2)**, bu yerda **n** - grafdagi uchlar soni.
Darhaqiqat, har bir uch ochilib, navbatga bir martagina joylashtirilgan, shuning uchun navbatning uchlari orasidagi sikl **n** martadan ko‘p bo‘lmagan vaqtda bajariladi.

Grafda o‘tish bo‘yi bo‘yicha qidiruv (DFS) algoritmi

- Grafda o‘tish bo‘yi bo‘yicha qidiruv (**DFS**) - bu graf uchlaridan o‘tishning rekursiv algoritmi.
- Agar bo‘yi bo‘yicha birmchi qidirish usuli nosimmetrik tarzda bajarilgan bo‘lsa (grafning uchlari darajalar bo‘yicha ko‘rib chiqilgan bo‘lsa), unda bu usul iloji boricha chuqurroq harakat qilishni o‘z ichiga oladi.



Algoritm qanday ishlashini aniq bir misol yordamida ko‘rib chiqamiz.



- Quyidagi yo‘naltirilmagan bog‘langan grafda jami 5 ta uch mavjud.
- Avval siz boshlang‘ich uchni tanlashingiz kerak.
- Qaysi uch tanlangan bo‘lsa ham, har qanday holatda ham graf to‘liq o‘rganib chiqiladi, chunki yuqorida aytib o‘tilganidek, bu bitta yo‘naltirilmagan bog‘langan graf.
- O‘tish 1 tugundan boshlasin, u holda qarab chiqilgan tugunlar ketma-ketligi tartibi quyidagicha bo‘ladi: 1 2 3 5 4.
- Agar ijro, masalan, 3 tugundan boshlangan bo‘lsa, u holda o‘tish tartibi boshqacha bo‘ladi: 3 2 1 5 4.

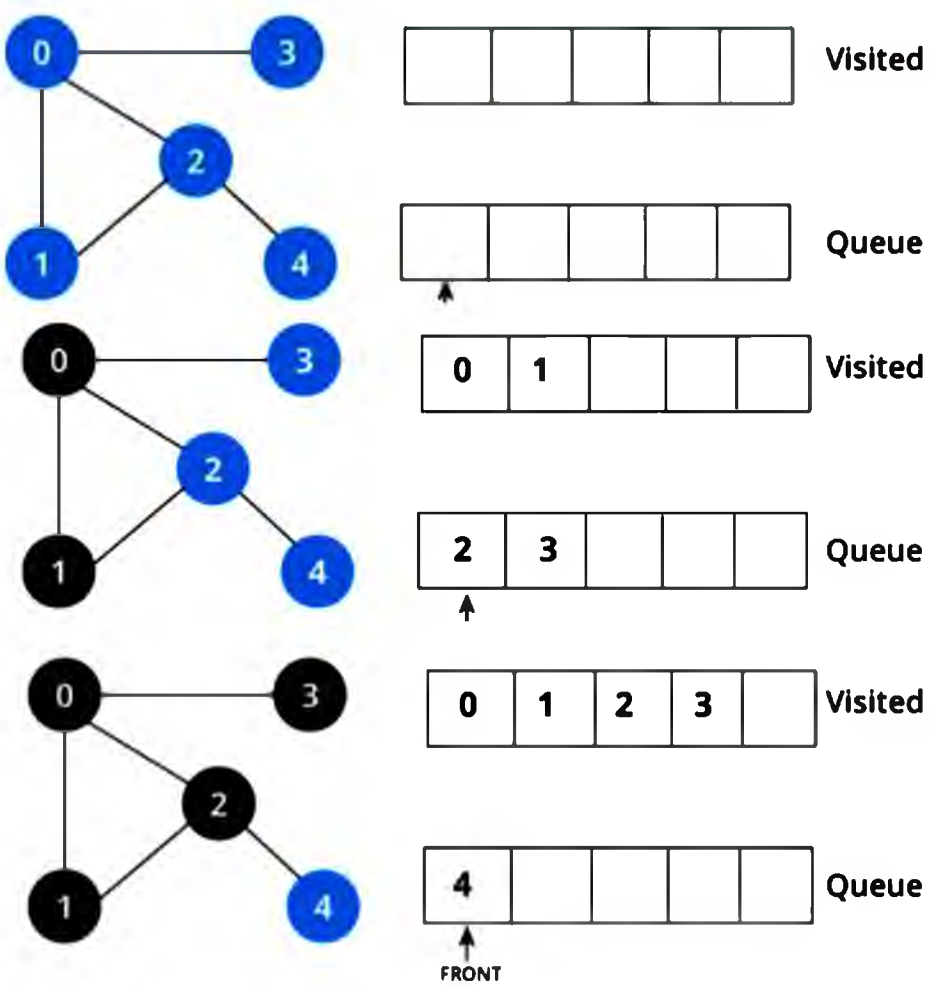
Algoritmning psevdokodi quyidagicha

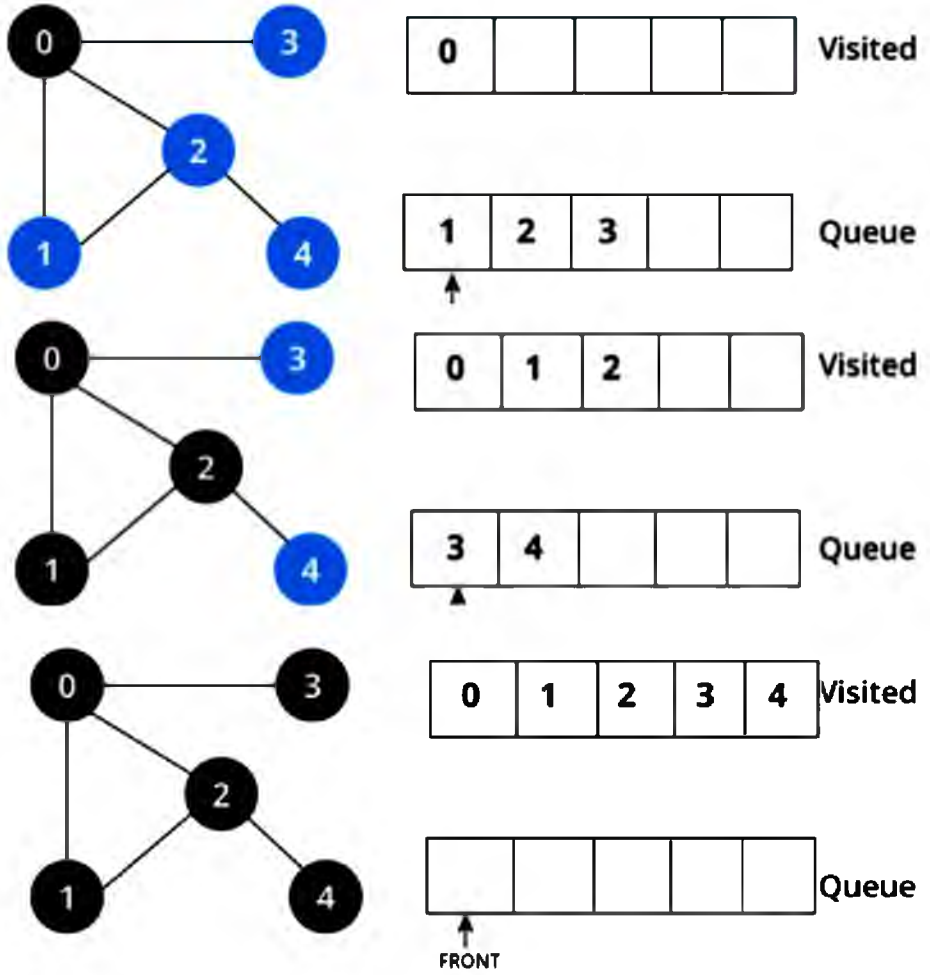
DFS funksiya sarlavhasi (st)
Chiqish (st)
visited[st] ← tashrif buyurgan;
r = 1 uchun n gacha
Agar (graph[st, r] ≠ 0) va (visited[r] tashrif buyurilmagan) bo‘lsa, u holda DFS (r)

- Bu **yerda DFS (deep-first search)** - bu funksiya nomi. Uning yagona parametri **st** - dasturning asosiy qismidan argument sifatida uzatiladigan boshlang‘ich uchdir.
- Mantiqiy qiymatlarni qabul qiladigan massivning har bir elementiga oldindan **false (yolg‘on, 0)** qiymat beriladi, ya’ni uchlarning har biri dastlab **tashrif buyurilmagan** deb yoziladi.

Misollar

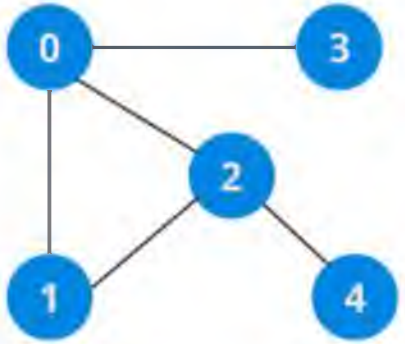
- **1-misol**
- BFS algoritmiga misol. Biz ikkita jadvalga ma‘lumotlarni joylashtirib boramiz
- Tashrif buyurilgan uchlar - **Visited** jadvaliga. Navbatda turgan uchlar esa - **Queue** jadvaliga

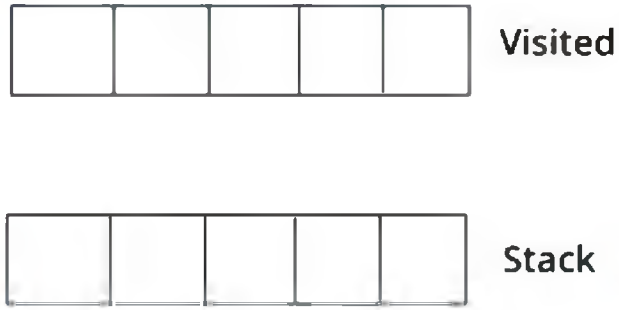


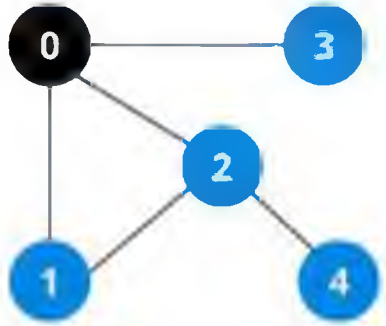


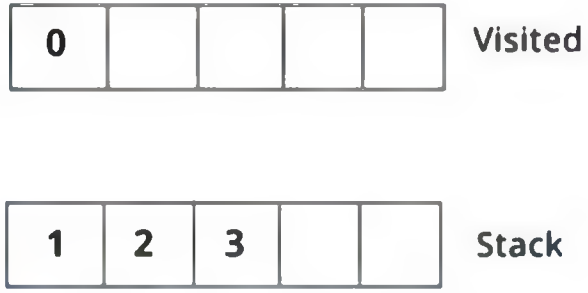
Misollar

- **2-misol**
- DFS algoritmining ishlashi. Biz 0 uchdan boshlaymiz, DFS algoritmi uni tashrif buyurilgan ro'yxatga qo'yishdan va barcha qo'shni uchlarni **Stek**ka joylashtirishdan boshlanadi.

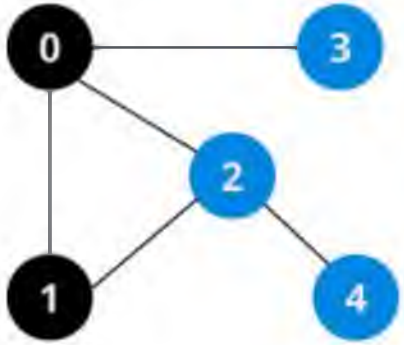


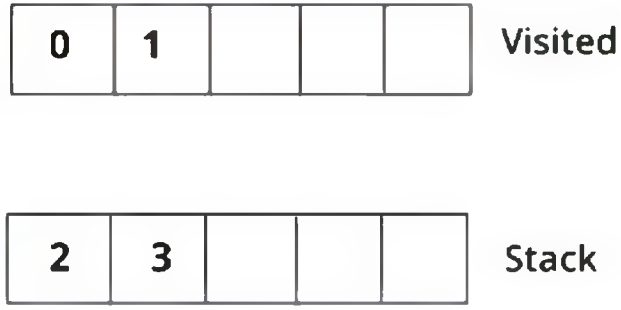






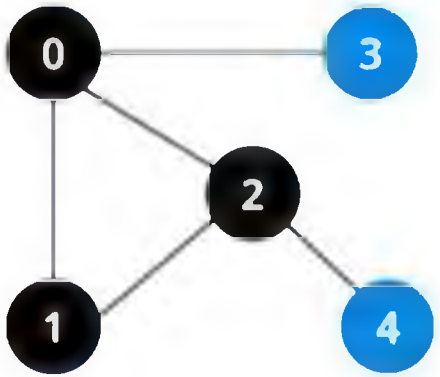
Keyin biz 1-uchning yuqori qismidagi elementga tashrif buyuramiz va qo'shni uchlarga o'tamiz. Biz allaqachon 0 ga tashrif buyurganimiz uchun, uning o'rniga 2 ga tashrif buyuramiz.





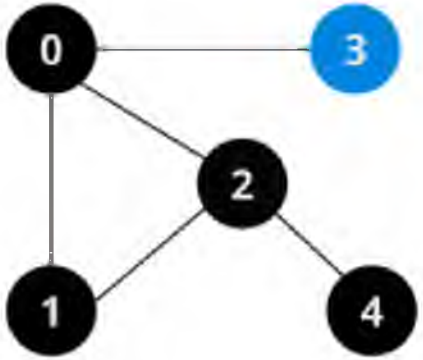
Misollar

- 2-uchda ko'rilmagan qo'shni 4-uch bor, shuning uchun biz uni to'plamning yuqori qismiga qo'shamiz va tashrif buyuramiz.



Visited

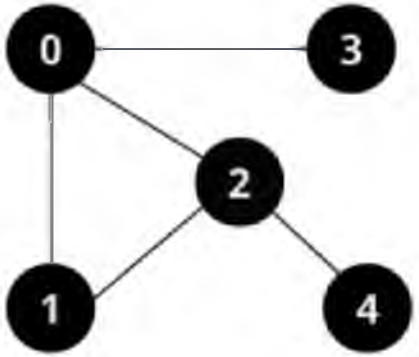
Stack



Visited

Stack

Oxirgi 3-bandga tashrif buyurganimizdan so'ng, uning ko'zga ko'rinmas qo'shni uchlar yo'q. Bu grafni birinchi chuqurlik birinchi o'tishini yakunlaydi.



Visited

Stack

Mavzu yuzasidan savollar:

1. Graflarda o‘tish algoritmlari qanday masala hisoblanadi?
2. BFS algoritmining ishlash prinsipi qanday?
3. DFS algoritmining ishlash prinsipi qanday?
4. Graflarda yana qanday o‘tish algoritmlari mavjud?
5. Yuqorida keltirilgan algoritmlarning murakkabligini baholang

Do you have any questions?