



## Breadth-first algoritmi va Dijkstra algoritmining farqlari.

*Saidmurotov Sulaymon Ollaberganovich, TATU Urganch filiali 2-bosqich talabasi*

+998886016030 [saidmurotov.sulaymon@mail.ru](mailto:saidmurotov.sulaymon@mail.ru)

*Ollaberganov Mirzoxid Ollabergan o'g'li Tatu urganch filiali 2-bosqich talabasi*

+998975170917 [mirzohidsher1859@gmail.com](mailto:mirzohidsher1859@gmail.com)

*Frunzayeva Dilfuza Axmad qizi TATU Urganch filiali 2-bosqich talabasi*

+998939291227 [dilfuzafrunzayeva@gmail.com](mailto:dilfuzafrunzayeva@gmail.com)

*Azizbek Sultonboyev Umarjon o'g'li Tatu urganch filiali 2-bosqich talabasi*

+998943198108 [azizbee@mail.ru](mailto:azizbee@mail.ru)

**Annotatsiya.** Ushbu maqolada Breadth-first va Dijkstra algoritmining farqlari haqida bayon etilgan. Bu ikkita algoritmi taqqoslash orqali ikkala algoritmi chuqurroq tadqiq etishga harakat qildik.

**Kalit so'zlar:** algoritim, grafika, navigatsiya, robot.

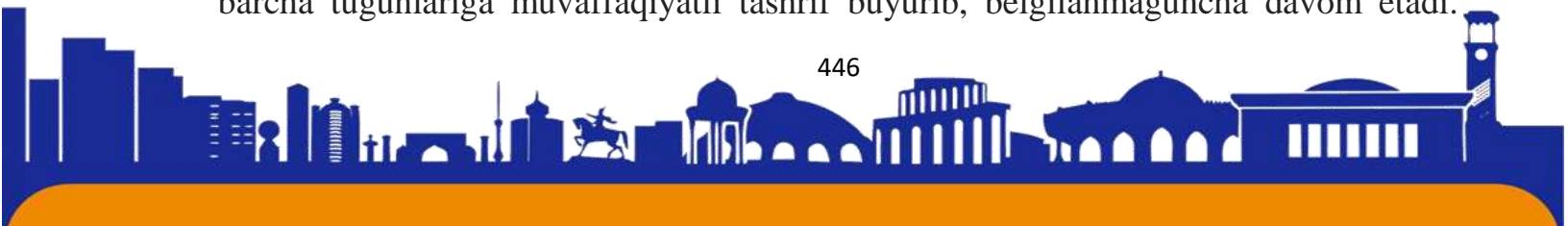
**Abstract.** This article explains the differences between Breadth-first and Dijkstra's algorithm. By comparing these two algorithms, we tried to explore both algorithms in depth.

**Key words:** algorithm, graphics, navigation, robot.

**Аннотация.** В этой статье объясняются различия между алгоритмом в ширину и алгоритмом Дейкстры. Сравнивая эти два алгоритма, мы попытались подробно изучить оба алгоритма.

**Ключевые слова:** алгоритм, графика, навигация, робот.

Breadth-first algoritmi bu eng qisqa yo'l topish uchun ishlatiladigan algoritim hisoblanadi. Bu qidiruv algoritim bo'lib, u ma'lumotlarning grafigini tuzish yoki daraxt yoki o'tish tuzilmalarini qidirish uchun ishlatiladi. BFS ning to'liq shakli kenglikdagi birinchi qidiruvdir. Algoritim grafikdagi barcha asosiy tugunlarga samarali tashrif buyuradi va to'g'ri kenglik bo'yicha belgilab beradi. Ushbu algoritim grafikdagi bitta tugunni (boshlang'ich yoki manba nuqtasi) tanlaydi va keyin tanlangan tugunga ulashgan barcha tugunlarga tashrif buyuradi. Esingizda bo'lsin, BFS ushbu tugunlarga birma-bir kirishadi. Algoritim boshlang'ich tugunga tashrif buyurib, belgilagandan so'ng, u eng yaqin ko'rilmagan tugunlarga o'tadi va ularni tahlil qiladi. Bir marta tashrif buyurganingizdan so'ng, barcha tugunlar belgilanadi. Ushbu takrorlashlar grafikning barcha tugunlariga muvaffaqiyatli tashrif buyurib, belgilanmaguncha davom etadi.



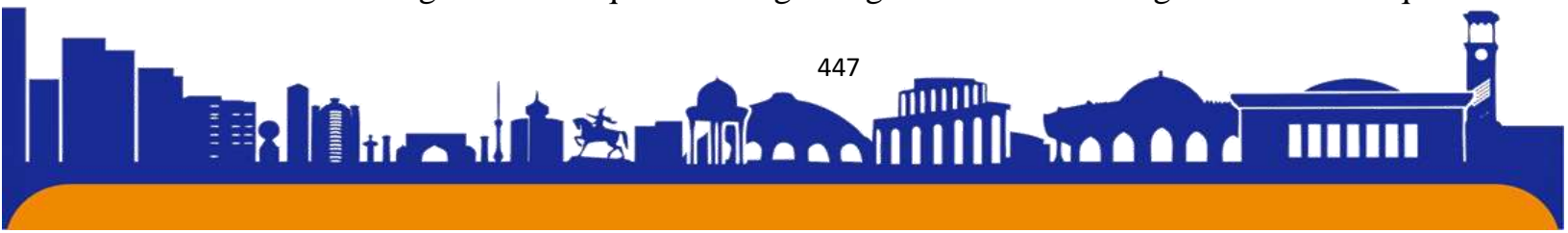


Grafik o'tish algoritmdan daraxtga o'xshash tuzilishdagi har bir tashrif buyurilmagan tugunga tashrif buyurishni, tekshirishni va/yoki yangilashni talab qiladi. Grafik o'tishlari grafikdagi tugunlarga tashrif buyurish tartibiga ko'ra tasniflanadi.

BFS algoritmi operatsiyani grafikning birinchi yoki boshlang'ich tugunidan boshlaydi va uni to'liq bosib o'tadi. Dastlabki tugunni muvaffaqiyatli kesib o'tgandan so'ng, grafikdagi keyingi o'tmagan cho'qqiga tashrif buyuriladi va belgilanadi. BFS algoritmining afzallik tomonlari shundaki berilgan masalaga albatta yechim topadi. Shuningdek, agarda bir nechta yechim mavjud bo'ladigan bo'lsa u minimal qadamlar ya'ni (eng qisqa) yo'lni topadi. Ammo, bu algoritmning kamchilik tomonlari ham mavjud. Xotira cheklovlari, ya'ni u keyingi bosqichga o'tish uchun o'zidagi barcha tugunlarni (ma'lumotlarni) saqlaydi. Agarda yechim uzoqda bo'ladigan bo'lsa u juda ko'p vaqt sarflaydi. Bu algoritmdan bugungi kunda juda ko'p joylarda foydalanishadi

- Navigatsiyada (Google maps, Waze, Yandex maps)
- Aloqa tarmoqlarida
- IP routing
- Ijtimoiy tarmoqlarda siz taniydigan odamlarni topishda
- Robot/dron uchun yo'l topishda

Dijkstra algoritmi yordamida siz grafikdagi tugunlar orasidagi eng qisqa yo'lni topishingiz mumkin. Xususan, siz tugundan ("manba tugun" deb ataladi) grafikdagi barcha boshqa tugunlarga eng qisqa yo'lni topishingiz mumkin, bu esa eng qisqa yo'l daraxtini hosil qiladi. Ushbu algoritm joriy joylashuv va maqsad o'rtasidagi eng qisqa yo'lni topish uchun GPS qurilmalarida qo'llaniladi. U sanoatda, xususan, modellashtirish tarmoqlarini talab qiladigan domenlarda keng qo'llanilishiga ega. Dijkstra algoritmi asosan siz tanlagan tugundan (manba tugun) boshlanadi va u ushbu tugun va grafikdagi barcha boshqa tugunlar orasidagi eng qisqa yo'lni topish uchun grafikni tahlil qiladi. Algoritm har bir tugundan manba tuguniga ma'lum bo'lgan eng qisqa masofani kuzatib boradi va agar u qisqaroq yo'l topsa, bu qiymatlarni yangilaydi. Algoritm manba tugun va boshqa tugun o'rtasidagi eng qisqa yo'lni topgandan so'ng, bu tugun "tashrif buyurilgan" deb belgilanadi va yo'lga qo'shiladi. Jarayon grafikdagi barcha tugunlar yo'lga qo'shilmaguncha davom etadi. Shunday qilib, biz manba tugunini har bir tugunga erishish uchun mumkin bo'lgan eng qisqa yo'l bo'ylab boshqa barcha tugunlarga bog'laydigan yo'lga egamiz. Bu algoritmning afzallik tomonlaridan biri. U bitta tugundan boshqa barcha tugunlarga va bitta manba tugunidan bitta maqsad





tuguniga eng qisqa yo'lni algoritmi to'xtatib, maqsad tugun uchun eng qisqa masofaga erishilgandan so'ng hisoblash uchun ishlatilishi mumkin. Ammo dijkstra algoritmining kamchilliklari ham bor masalan; Qayta ishlash jarayonida ko'p vaqt sarflaydigan noaniq qidiruvni amalga oshiradi.

N-node( tugun)

E- edge (qirra)

Log-logarifm

Breadth-first	Dijkstra
Bosh tugundan oxirgi tugungacha barcha tugunlarni ko'rib chiqadi	Har qadamda eng arzon tugunga o'tadi
Faqatgina vaznsiz (yoki bir hil vaznli) graflar bilan ishlaydi	Musbat vaznli va vaznsiz graflar bilan ishlaydi
Natija: "Eng qisqa" yo'l	Natija: "eng arzon" yo'l
Big O: $O(N+E)$	Big $O(N+E(\log N))$

### Foydalanilgan adabiyotlar

1. <https://www.geeksforgeeks.org>
2. <https://medium.com>
3. <https://www.freecodecamp.org>
4. <https://www.guru99.com>
5. <https://en.wikipedia.org>
6. <https://www.analyticssteps.com>

