UNIVERSITETI I PRISHTINES “HASAN PRISHTINA”

FAKULTETI I SHKENCAVE MATEMATIKE-NATYRORE

****

Drejtimi: Shkenca Kompjuterike

Lënda: Inteligjenca Artificiale

Tema: Algoritmi Breadth-first search

Punuan: Alketa Hajdari dhe Elda Meshi

**Hyrje**

Algoritmet e kërkimit janë një nga fushat më të rëndësishme të Inteligjencës Artificiale**.**Një komponent thelbësor i shkencës kompjuterike dhe inteligjencës artificiale janë algoritmet e kërkimit. Këto algoritme përdoren për të identifikuar një zgjidhje ose për të maksimizuar një rezultat të veçantë duke kërkuar nëpër grupe të mëdha të dhënash ose fusha të vështira problematike. Brute-force search, depth-first search, breadth-first search, heuristic search dhe metaheuristic search janë disa shembuj të llojeve të ndryshme të algoritmeve të kërkimit.Në këtë punim seminarik do të flasim për breadth-first search.

**Çka është breadth-first search?**

Breadth-First Search (BFS) bazohet në nyjet që kalojnë duke shtuar fqinjët e çdo nyje në radhën traversale duke filluar nga nyja e rrënjës. BFS për një grafik është i ngjashëm me atë të një peme, me përjashtim që grafikët mund të kenë cikle. Në kontrast me kërkimin e thellësisë së parë, të gjitha nyjet fqinje në një thellësi të dhënë hetohen përpara se të procedojnë në nivelin tjetër.

**Historia e BFS :**

Matematikani francez Charles Pierre Trémaux fillimisht shpiku metodën e kërkimit të parë në gjerësi (BFS), një teknikë themelore e përshkimit të grafikëve, në shekullin e 19-të. Megjithatë, gjatë viteve 1950 dhe 1960, shkencëtarët kompjuterikë popullarizuan algoritmin.

Kjo është një kronologji e shpejtë e si vijon:

1884: Pa shpikjen e Algoritmit të Trémaux të Charles Pierre Trémaux, një algoritëm për zgjidhjen e labirintit i bazuar në një variant të BFS. Megjithatë, algoritmi i tij nuk u krijua me kompjuterët në mendje.

1945: Në punën e tij të pabotuar mbi gjuhën e programimit Plankalkül, inxhinieri gjerman dhe pionieri kompjuterik Konrad Zuse përshkroi algoritmin BFS.

1950: Shkencëtarët e kompjuterave Claude Shannon dhe Edward F. Moore, duke punuar në teorinë e informacionit dhe teorinë e automatave, përkatësisht, krijuan në mënyrë të pavarur algoritmin BFS.

1959: Algoritmi BFS u botua për herë të parë në vitin 1959 nga shkencëtari holandez i kompjuterave E.W. Dijkstra në punimin e tij historik "Një shënim mbi dy probleme në lidhje me grafikët". Punimi i Dijkstra, i cili njihet gjerësisht si një nga botimet më me ndikim në shkencën kompjuterike, prezantoi BFS si një metodë për përcaktimin e rrugës më të shkurtër midis dy nyjeve në një grafik.

1962: Programi i parë kompjuterik që përdor algoritmin BFS u lëshua në vitin 1962 nga shkencëtari kompjuterik C. Y. Lee i Korporatës RAND. Algoritmi i Lee, i cili trajtoi problemin e labirintit, përfaqësoi një përparim të rëndësishëm në inteligjencën artificiale.

1972: Ronald Graham dhe bashkëpunëtorët e tij shpikën idenë e një radhe me shumë nivele në 1972 për të trajtuar grafikët me shumë nivele prioriteti, duke zgjeruar algoritmin BFS në skenarin me shumë nivele.

Që atëherë, një sërë aplikacionesh, duke përfshirë rrugëzimin e rrjetit, analizën e rrjeteve sociale, zvarritjen në ueb dhe inteligjencën artificiale, kanë përdorur gjerësisht algoritmin BFS.

**Aplikacionet BFS**

Për shkak të fleksibilitetit të algoritmit, Breadth-First Search është mjaft i dobishëm në botën reale. Këto janë disa prej tyre:

* Në një rrjet peer-to-peer, zbulohen nyjet peer. Shumica e klientëve të torrentit, si BitTorrent, uTorrent dhe qBittorent, përdorin këtë proces për të gjetur "seeds" dhe "peers" në rrjet.
* Indeksi është ndërtuar duke përdorur teknikat e kalimit të grafikut në zvarritjen e uebit.

Procedura fillon me faqen burimore si nyje rrënjësore dhe shkon deri te të gjitha faqet dytësore që janë të lidhura me faqen burimore (dhe ky proces vazhdon). Për shkak të thellësisë së reduktuar të pemës së rekursionit, Breadth-First Search ka një avantazh të qenësishëm këtu.

* Përdorimi i sistemeve të navigimit GPS duke përdorur GPS, kryen një kërkim të gjerë për të gjetur vendndodhjen e afërt.
* Teknika e Cheney, e cila përdor konceptin e kërkimit të parë në gjerësi, përdoret për të mbledhur mbeturina.

**Algoritmi:**

Hapat e përfshirë në algoritmin BFS për të eksploruar një grafik janë dhënë si më poshtë:

**Step1:** SET STATUS = 1 (gjendje e gatshme) për secilën nyje në G

**Step 2:** Vendosni nyjen fillestare A dhe vendosni STATUSIN e saj = 2 (gjendja e pritjes)

**Step 3:** Përsëritni hapat 4 dhe 5 derisa QUEUE të jetë bosh

**Step 4:** Zhvendosni një nyje N. Përpunoni atë dhe vendosni STATUSIN e saj = 3 (gjendja e përpunuar).

**Step 5:** Vendosni të gjithë fqinjët e N që janë në gjendje gatishmërie (STATUSI i të cilëve = 1) dhe vendosni

STATUSI i tyre = 2

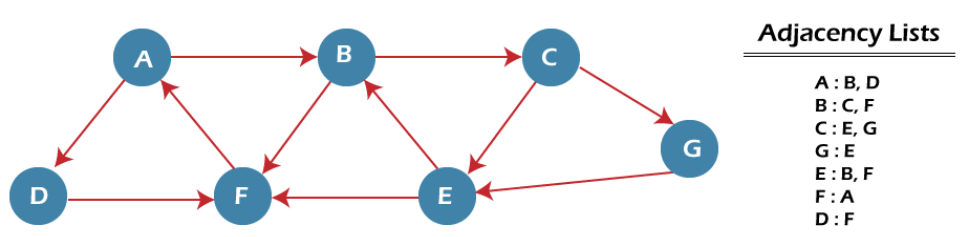
(gjendja e pritjes)

[FUNDI I LAKUT]

**Step 6:** DALJE

**Shembull i algoritmit BFS**

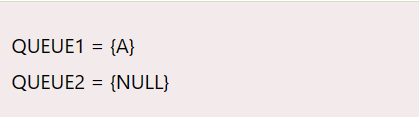
Tani, le të kuptojmë funksionimin e algoritmit BFS duke përdorur një shembull. Në shembullin e dhënë më poshtë, ekziston një grafik i drejtuar me 7 kulme



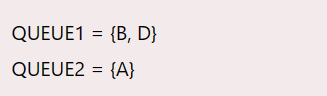
Në grafikun e mësipërm, shtegu minimal 'P' mund të gjendet duke përdorur BFS që do të fillojë nga nyja A dhe do të përfundojë në nyjen E. Algoritmi përdor dy radhë, përkatësisht QUEUE1 dhe QUEUE2. QUEUE1 mban të gjitha nyjet që do të përpunohen, ndërsa QUEUE2 mban të gjitha nyjet që përpunohen dhe fshihen nga QUEUE1.

Tani, le të fillojmë të shqyrtojmë grafikun duke filluar nga nyja A.

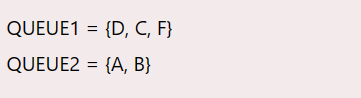
**Step 1 -**QUEUE1 = {A}



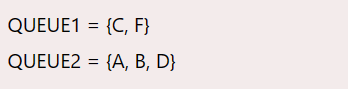
**Step2-**Tani, fshini nyjen A nga radha1 dhe shtoni atë në radhën2. Futni të gjithë fqinjët e nyjës A në radhën1.



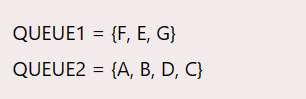
**Step3:** Tani, fshini nyjen B nga radha1 dhe shtoni atë në radhën2. Futni të gjithë fqinjët e nyjës B në radhën1



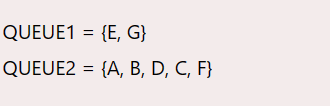
**Step 4-**Tani, fshini nyjen D nga radha1 dhe shtoni atë në radhën2. Futni të gjithë fqinjët e nyjës D në radhën1. Fqinji i vetëm i nyjës D është F pasi është futur tashmë, kështu që nuk do të futet më**.**

****

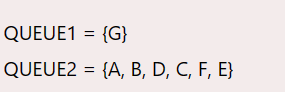
**Step 5-**Fshini nyjen C nga radha1 dhe shtoni atë në radhën2. Futni të gjithë fqinjët e nyjës C në radhën1**.**

****

**Step 5-**Fshini nyjen F nga radha1 dhe shtoni atë në radhën2. Futni të gjithë fqinjët e nyjës F në radhën1. Meqenëse të gjithë fqinjët e nyjës F janë tashmë të pranishëm, ne nuk do t'i fusim më.



**Step 6-**Fshi nyjen E nga radha1. Meqenëse të gjithë fqinjët e saj tashmë janë shtuar, kështu që ne nuk do t'i fusim më ato. Tani, të gjitha nyjet janë vizituar dhe nyja e synuar E gjendet në radhën2.



**Kompleksiteti i algoritmit BFS**

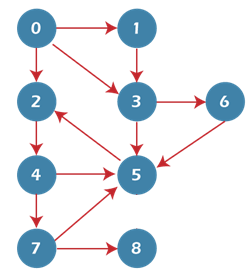
Kompleksiteti kohor i BFS varet nga struktura e të dhënave e përdorur për të përfaqësuar grafikun. Kompleksiteti kohor i algoritmit BFS është O(V+E), pasi në rastin më të keq, algoritmi BFS eksploron çdo nyje dhe skaj. Në një grafik, numri i kulmeve është O(V), ndërsa numri i skajeve është O(E).

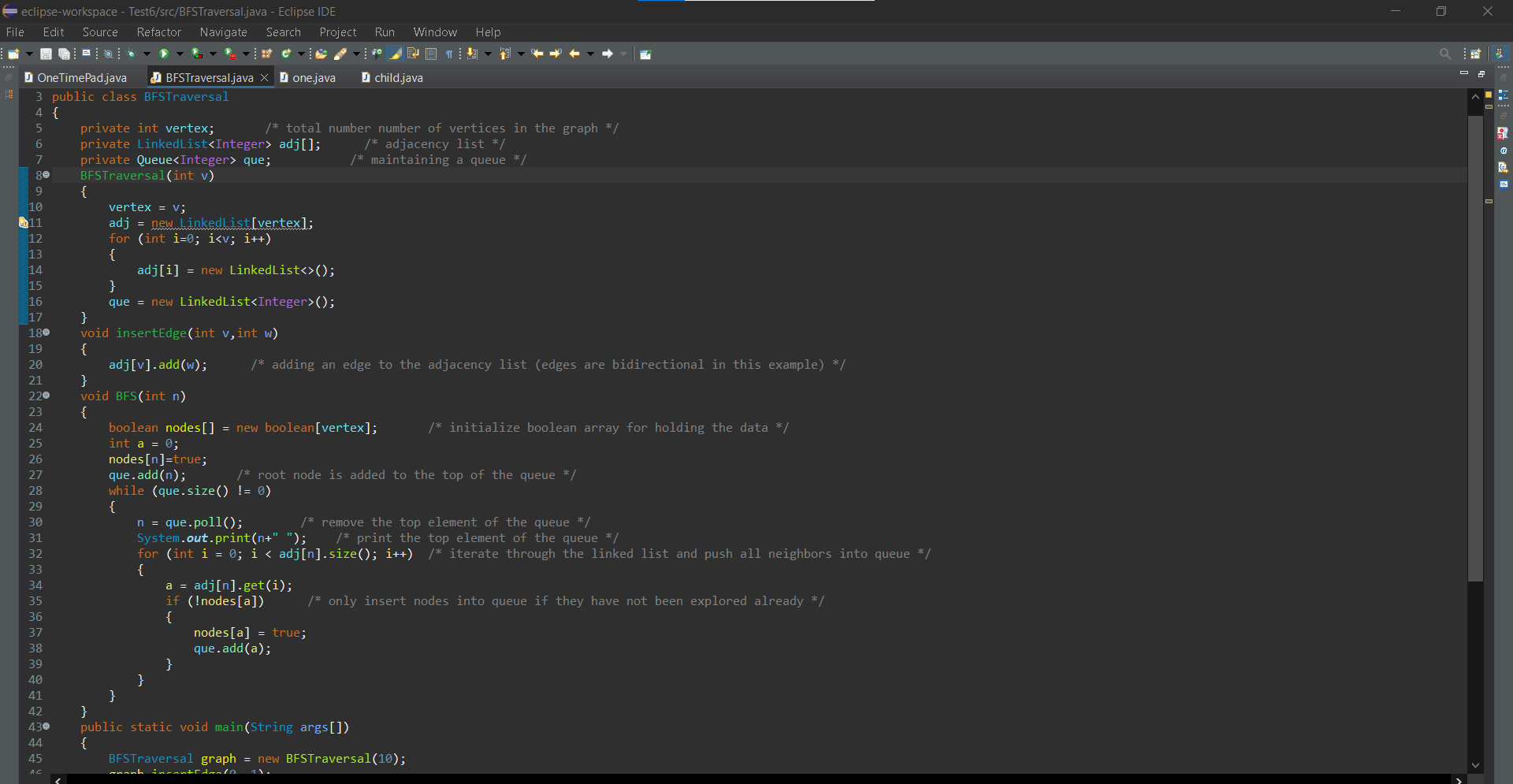
Kompleksiteti hapësinor i BFS mund të shprehet si O(V), ku V është numri i kulmeve.

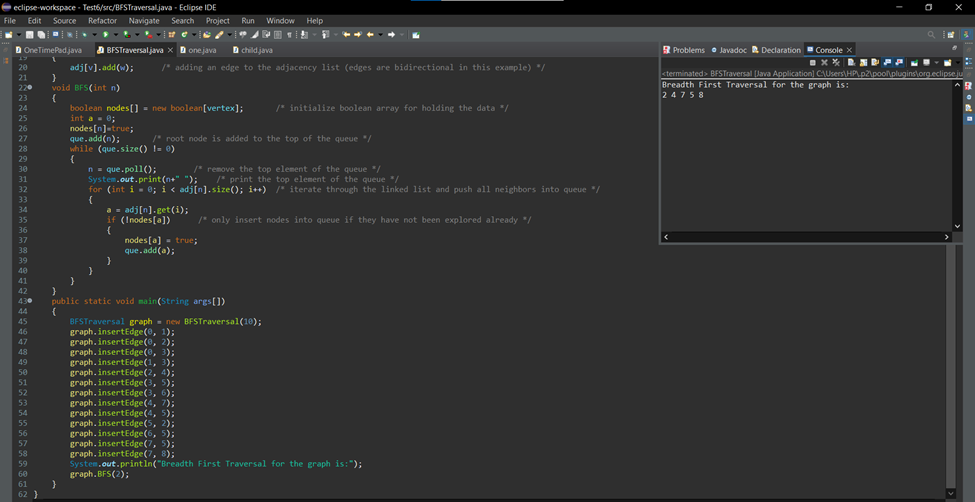
**Implentimi(zbatimi) i algoritmit BFS**

Tani, le të shohim zbatimin e algoritmit BFS në java.

Në këtë kod, ne po përdorim listën e fqinjësisë për të përfaqësuar grafikun tonë. Zbatimi i algoritmit Breadth-First Search në Java e bën shumë më të lehtë trajtimin e listës së afërsisë pasi ne duhet të udhëtojmë nëpër listën e nyjeve të bashkangjitura në secilën nyje pasi nyja të hiqet nga kreu (ose fillimi) i radhës.

Në këtë shembull, grafiku që ne po përdorim për të demonstruar kodin është dhënë si më poshtë 





**Konkluzioni**

Në këtë punim seminarik, ne kemi diskutuar teknikën e kërkimit Breadth-first së bashku me shembullin, kompleksitetin dhe zbatimin e saj në gjuhën e programimit java. Këtu, ne kemi parë gjithashtu aplikacionet e jetës reale të BFS që i tregojnë atij algoritmin e rëndësishëm të strukturës së të dhënave.