SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO - MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Kvantno računanje

MINIMALNO RAZAPINJUĆE STABLO

SEMINAR

Fran Vojković
Alen Živković

veljača, 2020.

Sadržaj

1 Uvod

Primarni cilj teorije kvantnog računanja je odrediti u kojim slučajevima kvantna računala omogućuju brže izvršavanje algoritama od klasičnih računala. U seminaru promatramo problem nalaženja *minimalnog razapinjućeg stabla (MST* [1]*)* opisanog u poglavlju 2, gdje smo naveli dva klasična algoritma, *Primov* i *Kruskalov* algoritam, za navedeni problem. Poglavlje 3 sadrži terminologiju potrebnu za formulaciju i analizu složenosti kvantnog algoritma opisanog u poglavlju 4. Kvantni algoritam temeljimo na *Boruvka* algoritmu. Također, u poglavlju 4.2 promatramo problem povezanosti grafa koji je poseban slučaj problema MST-a u kojem su nam sve težine jednake te nas zanima postoji li podgraf u kojem postoji put između svaka dva vrha.

[1] Minimum spanning tree

2 Standardni algoritmi za MST

Neka je $G = (V, E)$ povezan neusmjeren graf sa skupom vrhova V i skupom bridova E. Svaki brid ima nenegativnu duljinu.

Problem MST: Treba naći podskup T grafa G tako da svi vrhovi ostanu povezani samo sa bridovima iz T i da je zbroj duljina (cijene) bridova u T najmanji moguć. Uočimo da je podgraf (V, T) grafa G *stablo*, tj. povezan graf bez ciklusa. Taj graf zovemo minimalno razapinjuće stablo grafa G.

Predstavit ćemo dva standardna pohlepna algoritma za rješavanje navedenog problema. Uvodimo sljedeću terminologiju koja će nam trebati za dokaz korektnosti algoritama.

Definicija 1 *Za skup bridova kažemo da je* ***dopustiv*** *ako ne sadrži ciklus. Dopustiv skup bridova je* ***obećavajući*** *ako se može dopuniti do optimalnog rješenja. Brid e* ***dira*** *skup vrhova, ako je točno jedan kraj brida u tom skupu vrhova.*

Lema 1 *Neka je $G = (V, E)$ povezan neusmjeren graf sa zadanom duljinom bridova. Neka je $B \subset V$ te $T \subseteq E$ obećavajući skup bridova, takav da ni jedan brid iz T ne dira B. Neka je e najkraći brid koji dira B.*

Tada je i skup $T \cup \{e\}$ obećavajući.

Dokaz: T je obećavajući po pretpostavci pa postoji minimalno razapinjuće stablo U grafa G takvo da je $T \subseteq U$.

Ako je $e \in U$ tvrdnja vrijedi. U suprotnom, $e \notin U$ kada brid e dodamo u U onda nastaje jedan ciklus, jer je U stablo. Kako e dira B, mora postojati bar jos jedan brid $\tilde{e}$ koji također dira B (u suprotnom se ciklus ne zatvara). Ako izbacimo $\tilde{e}$, ciklus nestaje i dobivamo novo stablo $\tilde{U}$ koje razapinje graf G. No po pretpostavci, duljina brida e ne prelazi duljinu brida $\tilde{e}$ pa ukupna duljina bridova u $\tilde{U}$ ne prelazi ukupnu duljinu bridova u U. Tada je i $\tilde{U}$ minimalno razapinjuće stablo za graf G i sadrži brid e. Mora vrijedit $T \subseteq \tilde{U}$ jer e ne može biti u T ($\tilde{e}$ dira B, a niti jedan brid iz T, po pretpostavci , ne dira B). Slijedi da je i $T \cup \{e\}$ obećavajući u $\tilde{U}$.

■

2.1 Primov algoritam

Inicijaliziramo stablo B s jednim proizvoljnim vrhom te skupom bridova T koji je prazan. U svakom koraku dodajemo jedan novi brid stablu te algoritam staje kada povežemo sve vrhove. U jednom koraku, algoritam nalazi najkraći brid $\{u, v\}$ takav da je

$$(u \in V \setminus B) \wedge (v \in B) \tag{1}$$

te dodaje u u skup B i brid $\{u, v\}$ u skup T. Iz konstrukcije slijedi da bridovi u T u svakom trenutku predstavljaju minimalno razapinjuće stablo za vrhove iz B. Postupak se nastavlja dok je $B \neq V$.

Složenost *Primovog algoritma* je $\mathcal{O}(n^2)$.

2.2 Kruskalov algoritam

Neka je skup bridova $T = \emptyset$, tj. svaki vrh grafa G čini zasebnu komponentu povezanosti.

1. Sortiramo bridove E uzlazno po duljini.
2. Uzimamo najkraći brid e iz E.
   (a) Ako spaja dva vrha u različitim komponentama povezanosti, dodajemo e u T.
   (b) Ako spaja dva vrha u istoj komponenti povezanosti, odbacujemo brid e.
3. STOP ako je preostala jedna komponenta povezanosti.

Za povezan neusmjeren graf *Kruskalov algoritam* radi korektno te je struktura pogodna za implementaciju algoritma *struktura disjunktnih skupova.* Složenost *Kruskalovog algoritma* je $\mathcal{O}(mlog(n))$, gdje je m broj bridova u grafu.

Ako je graf G gust, odnosno ima mnogo bridova, slijedi $m \approx \frac{n(n-1)}{2}$ pa je složenost algoritma $\mathcal{O}(n^2 log(n))$, u slučaju da je graf G rijedak vrijedi $m \approx n - 1$, pa je složenost algoritma $\mathcal{O}(nlog(n))$, te je *Kruskalov* algoritam, bitno brži od *Primovog* algoritma.

3 Terminologija

Neka oznaka $[n]$ predstavlja skup $\{0, 1, ..., n-1\}$. Graf je uređeni par $G = (V, E)$ gdje je V skup čiji se elementi zovu vrhovi, a $E \subseteq V \times V$ skup čiji se elementi zovu bridovi. Promatramo dva modela za usmjerene grafove.

Definicija 2 ***Model matrice susjedstva****, u kojem je graf dan kao matrica susjedstva $M \in \{0,1\}^{n\times n}$, gdje $M_{ij} = 1$ ako i samo ako $(v_i, v_j) \in V$.*

Primjer:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Iz ove se matrice susjedstva da iščitati da su u našem grafu prisutni bridovi (v_0, v_1), (v_1, v_0), (v_1, v_1) i (v_2, v_2).

Definicija 3 ***Vektorski model****, gdje za svaki vrh v_i imamo broj njegovih susjeda $d_1, d_2, ..., d_n$ te polje njegovih susjeda $f_i[d_i] \to [n]$. Dakle, $f_i(j)$ vraća j-tog susjeda vrha v_i po nekom proizvoljnom, ali fiksnom, numeriranju susjeda. Smatramo da su vrijednosti d_i dane unosom za svaki i pa računamo samo pozive funkcija f_i.*

Također, funkcije zadovoljavaju

$$(\forall i \in [n])(j, j' \in [d_i]) \text{ t.d. } j \neq j' : f_i(j) \neq f_i(j')$$

Dakle, graf nije multigraf, tj. nema više od jednog brida između bilo koja dva vrha.
Jedan od alata koji će nam biti potreban u realizaciji kvantnog algoritma za minimalno razapinjuće stablo je algoritam za pronalaženje d najmanjih vrijednosti različitog tipa. Problem glasi ovako:

***Problem nalaželja d najmanjih (različitih) vrijednosti*:**
Imamo dvije funkcije f,g i cijeli broj d' te želimo pronaći cijeli broj $d = min\{d', e\}$ i podskup $I \subseteq [N]$ kardinalnosti d takav da $g(i) \neq g(i')$ za sve različite $i, i' \in I$ i da za sve $j \in [N] \setminus I$ i $i \in I$, ako $f(j) < f(i)$ onda $f(i') \leq f(j)$ za neki $i' \in I$ takav da $g(i') = g(j)$.

Sljedeći teorem daje nam donju ogradu za složenost algoritma za *problem nalaželja d najmanjih (različitih) vrijednosti.*

Teorem 1 *Donja granica složenosti za* ***Problem nalaženja d najmanjih (različitih) vrijednosti*** *je* $\Omega(\sqrt{dN})$.

Dokaz: Za k paran i d neparan promatramo matrice $C \in M_{d,k}(\{0,1\})$ koje u svakom retku imaju točno jednu 0. Takve matrice opisuju funkciju $f : \{0,1,...,N-1\} \to \{0,1\}$ tako da

$$\forall i \in \{0,1,...,d-1\}, j \in \{0,1,...,k-1\} \ , \ f(id+j) = C_{i,j}$$

Neka je $g : \{0,1,...,N-1\} \to \{0,1,...,d\}$ takva da

$$g(id+j) = \begin{cases} i & \text{, ako } f(id+j) = 0 \\ d & \text{, inače} \end{cases}$$

Sada je problem nalaženja $d+1$ najmanjih vrjdenosti ekvivalentan nalaženju pozicija d nula u matrici C. Neka je

$$X := \{A \in M_{d,k}(\{0,1\}) : \text{ točno } \lfloor d/2 \rfloor \text{ redaka ima 0 u prvih } k/2 \text{ stupaca } \}$$

$$Y := \{A \in M_{d,k}(\{0,1\}) : \text{ točno } \lceil d/2 \rceil \text{ redaka ima 0 u prvih } k/2 \text{ stupaca } \}$$

Kažemo da je matrica $A \in X$ u relaciji sa matricom $B \in Y$ ako i samo ako se razlikuju na točno dva mjesta.

Iz navedenog slijedi da $\exists i \in \{0,1,...,d-1\}$, $0 \leq j < k/2 \leq \tilde{j} < k$ t.d. $A_{i,j} = B_{i,\tilde{j}} = 1 \wedge A_{i,\tilde{j}} = B_{i,j} = 0$. Npr.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} B = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Broj matrica koje su u relaciji sa fiksiranom matricom je najmanje $m = \tilde{m} = \lceil d/2 \rceil k/2$. Za fiksirane i, j broj matrica u relaciji sa M koje se razlikuju u i, j je $k/2$ ako $M_{i,j} = 0$ i 1 ako $M_{i,j} = 1$. Pa za $l_{max} = k/2$, $\Omega(\sqrt{m\tilde{m}/l_{max}})$ daje traženu donju granicu složenosti.

■

3.1 Groverov algoritam

Prethodni problem nalaženja najmanjih vrijednosti možemo rijesšiti *Grover*ovim algoritmom pretraživanja. Za navedeni algoritam potrebno je konstruirati kvantnu proročicu koja će otkriti tražene vrijednosti. Implementaciju kvantne proročice možemo opisati kvantnim sklopom koji okreće pomočni qubit q ako sljedeća funkcija iznosi 1.

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{, ako } \boldsymbol{x} = \tilde{x} \\ 0 & \text{, ako } \boldsymbol{x} \neq \tilde{x} \end{cases}$$

pri čemu je $\tilde{x}$ traženi element, a $x = (x_1, x_2)$ binarna varijabla. Pri čemu koristimo *Toffoli*jeva vrata koja uzimaju tri qubita te vraćaju prva dva qubita nepromjenjena, a treći qubit obrnemo ako su prva dva qubita 1. Implementacijom *Toffoli*jevih vrata dobijemo traženu kvantnu proročicu gdje nam prva dva qubita predstavlja ulazne parametre, a treći qubit, q, je pomoćni.

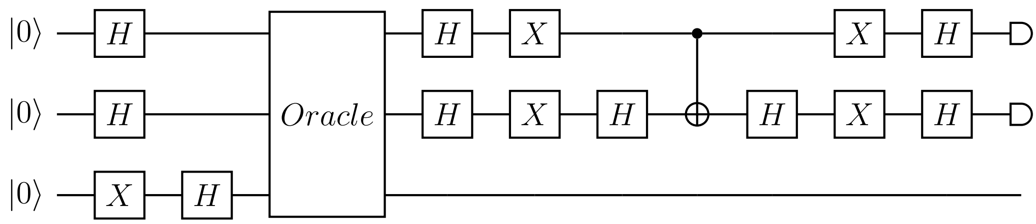
Ponašanje naše proročice općenito izgleda ovako $|\boldsymbol{x}\rangle |q\rangle \rightarrow |\boldsymbol{x}\rangle |f(\boldsymbol{x}) \oplus q\rangle$. Groverov operator je ključ za rad ovog algoritma. Definiramo ga ovako

$$G = (2\,|\psi\rangle \langle\psi| - I)O$$

gdje je ψ uniformna superpozicija svih stanja, a O naša proročica. Djelovanje $(2\,|\psi\rangle \langle\psi| - I)$ na proizvoljno stanje je

$$(2\,|\psi\rangle \langle\psi| - I) \sum_i a_i\,|i\rangle = \sum_i (2\langle a\rangle - a_i)\,|i\rangle \tag{2}$$

gdje je $\langle a\rangle = \frac{\sum_i a_i}{\sum_i 1}$ i sume idu po svim mogućim stanjima. Kako bi Groverov operator uspješno obavio pretragu, $|x\rangle |q\rangle$ moramo pravilno inicijalizirati. Primijenimo Hadamardovu transformaciju ($H^{\otimes n}$) na $|x\rangle$ i Paulijevu X transformaciju pa Hadamardovu transformaciju na $|q\rangle$. Ovo ostavlja $|x\rangle$ u uniformnoj superpoziciji svih stanja $|\psi\rangle$, a $|q\rangle$ u stanju $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$.



Shematski dijagram Groverovog algoritma. Do proročice je proces inicijalizacije, a sve nakon nje (uključujući nju) je Groverov operator. Primijetimo da samo jednom primijenjujemo Groverov operator. Ovo je dostatno kad x sadrži samo dva bita, ali za složenije se probleme primijenjuje više puta.

Koristeći jednadžbu (2) sada vidimo kako Groverov operator radi. Primijenom proročice na $|x^*\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ obrnemo amplitudu tog stanja.

$$O\,|x^*\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \rightarrow |x^*\rangle \frac{|f(x^*) \oplus 0\rangle - |f(x^*) \oplus 1\rangle}{\sqrt{2}} = |x^*\rangle \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -\,|x^*\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Sličnim argumentom pokažemo da sva ostala stanja ostaju nepromjenjena proročicom. Kada ovo kombiniramo s jednadžbom (2), postaje jasno zašto Groverov operator uspješno provodi pretragu. U općenitom slučaju Groverov bi operator trebalo primijeniti $\lceil \frac{\pi 2^{n/2}}{4} \rceil$ puta.

4 Kvantni algoritam za minimalno razapinjuće stablo

Neka je $G = (V, E)$ neusmjeren težinski graf. Želimo naći podgraf grafa G bez ciklusa s maksimalnom kardinalnosti E tako da podgraf ima minimalnu ukupnu težinu.

U poglavlju 2 opisali smo *Primov* i *Kruskalov* akgoritam, također postoji i *Boruvka* algoritam koji sada koristimo. U najgorem slučaju, *Boruvka* algoritam ima $log(n)$ iteracija. Inicijaliziramo ga sa n razapinjućih stabala, tako da svako stablo sadrži jedan vrh iz E. U svakoj iteraciji algoritam nalazi brid s minimalnom težinom i dodaje ga u stabla, odnosno spaja ih u manje većih stabala. Nakon najvise $log(n)$ iteracija preostaje najviše jedno stablo koje predstavlja traženo minimalno razapinjuće stablo.

Tvrdnja 1 *Neka je $U \subset V$, povezanog grafa $G = (V, E)$, te neka je e minimalni brid po težini od $(U \times \tilde{U}) \cap E$.*
Tada postoji minimalno razapinjuće stablo koje sadrži brid e.

Modificiramo standardni *Boruvka* algoritam tako da kvantna verzija *Boruvka* algoritma ima najmanju vjerojatnost greške. Postavimo ga tako da je u l-toj iteraciji vjerojatnost greške najviše $\frac{1}{2^{l+2}}$ pa je ukupna greška najviše 0.25.

Kvantni algoritam *Boruvka*:

1. Neka je $T_1, T_2, ..., T_k$ razapinjuća šuma. Inicijalno neka je $k = n$ i neka svaki T_i sadrži točno jedan vrh.
2. Neka je $l = 0$.
3. Dok $k \neq 1$:
   (a) Povećaj l za jedan.
   (b) Naďi bridove $e_1, e_2, ..., e_k$ takve da je brid e_j minimalan po težini koji napušta T_j.
   Prekidamo ako je broj upita $(l + 2)c\sqrt{km}$ za konstantu c.
   (c) dodaj bridove $e_j^{'}$ u stabla, spajanjem u veća stabla.
4. Vrati razapinjuće stablo T_1.

Kako bi našli minimalne bridove u koraku (3b) koristimo sljedeće funkcije. U vektorskom modelu svaki vrh (u, v) se pojavljuje dva puta , odnosno u se pojavljuje kao susjedni vrh od v te v kao susjedni od u. Numeriramo usmjerene bridove od 0 do $2m-1$. Neka je $f : [0, 1, ..., 2m-1] \to \mathbb{N}^*$ funkcija koja pridružuje svim usmjerenim bridovima (u, v) njihove težine ako u i v pripadaju različitim stablima od trenutnog razapinjućeg, inače pridruži ∞.

Neka je $g : [0, 1, ..., 2m - 1] \to [0, 1, ..., k - 1]$ funkcija koja pridružuje usmjerenom bridu (u, v) indeks j stabla T_j koje sadrži u. Primijenimo algoritam za *nalaženje k najmanjih vrijednosti različitih tipova*, prekidajući ga nakon $(l+2)c\sqrt{km}$ iteracija kako bi dobili grešku vjerojatnosti najviše $1/2^{l+2}$.

Teorem 2 *Neka je dan neusmjeren graf sa težinama bridova. Navedenim algoritmom dobivamo razapinjuće stablo koje je minimalno sa vjerojatnošću najmanje* $1/4$. *Algoritam koristi* $O(\sqrt{nm})$ *upita u vektorskom modelu i* $O(n^{3/2})$ *upita u matričnom modelu.*

Dokaz: [za vektorski model] Na početku l-te iteracije broj stabla k je najviše $n/2^{l-1}$ iz čega slijedi da koristi najviše $(l+2)c\sqrt{nm/2^{l-1}}$ upita. Sumiramo po svim iteracijama pa je ukupan broj upita $O(\sqrt{nm})$. l-ta iteracija predstavlja grešku sa vjerojatnosti $\frac{1}{2^{l+2}}$ pa je ukupna vjerojatnost greške $\leq \frac{1}{4}$.
■

Napomena 1 *Dokaz vrijedi i za matrični model ako matricu promatramo kao poseban slučaj vektorskog modela s* $m = n(n-1)$ *bridova.*

Teorem 3 *Donja granica složenosti algoritma za traženje minimalnog razapinjućeg stabla je* $\Omega(\sqrt{nm})$.

Dokaz:

Slično kao i u teoremu 1 za nalaženje d najmanjih vrijednosti. Neka je $m = k(n+1)$ za $k \in \mathbb{N}$ te $M \in M_{n,k}(\mathbb{R}^+)$. Iz teorema 1, za $d = n, N = kn$, slijedi da je donja granica za nalaženje d najmanjih vrijednosti u svakom retku $\Omega(\sqrt{kn^2})$. Konstruiramo težinski graf G iz M na sljedeći način. Vrhovi grafa G su $V = \{s, v_1, ..., v_k, u_1, ..., u_n\}$ gdje je s korijen stabla, a $v_1, ..., v_k$ njegova djeca u prvoj razini, itd. Bridovi oblika (s, v_i) imaju težinu 0, a (v_i, u_j) težinu $M_{i,j}$. Očito MST sadrži bridove težine 0 koji spajaju s sa v_i.
■

4.1 Implementacija

Koristimo *Grover*ov algoritam u koraku 3b *Boruvka* algoritma. Neka je $N = 2^n$ te $f : \{0, 1, ..., N-1\} \to \{0, 1\}$ definirana kao

$$f(i) = \begin{cases} 1 & \text{, ako tražimo } i\text{-ti element} \\ 0 & \text{, inače } \tilde{x} \end{cases}$$

Definiramo Kvantnu proročicu koja djeluje na n qubita na sljedeći način.

$$U_f(|x\rangle) = (-1)^{f(x)} |x\rangle \tag{3}$$

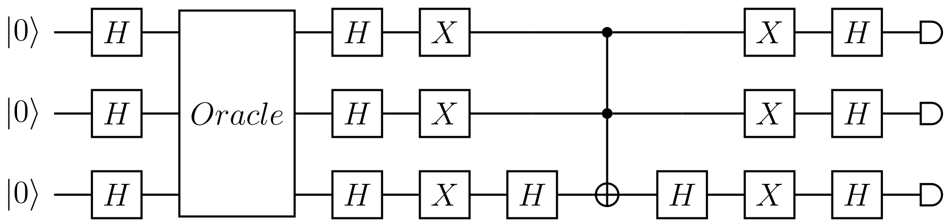
Idealno bi blo kada bi implementirali f kao kvantni sklop koji evaluira u 1 nakon kratkog vremena, no to ne možemo napraviti direktno implementirajući f kako moramo provjeriti N elemenata u listi. Pa implementiramo algoritam na sljedeći način

1. Inicijaliziramo j na proizvoljno mjesto u listi.

2. Ponavljamo:

   (a) Nađi $i \leq j$ u listi.

   (b) Postavi $j := i$.



Shematski prikaz Groverovog algoritma

4.2 Povezanost

Problem povezanosti grafa G predstavlja posebni slučaj minimalnog razapinjućeg stabla. Neka je dan neusmjeren graf G sa bridovoma jednakih težina, odnosno bez težina. Predstavit ćemo pohlepni kvantni algoritam za nalaženje MST-a u grafu G sa navedenim svojstvima. Za matrični model algoritam će u najgorem slučaju imati složenost $O(n^{3/2})$, a za vektorski model složenost će biti $O(n)$.

Teorem 4 *Neka je M matrica susjedstva grafa G. Sljedeći algoritam vraća MST nakon najviše $n^{3/2}$ iteracija ako je G povezan graf, inače algoritam ne staje.*

Algoritam za matrični model:

1. Inicijaliziramo skup bridova $A = \emptyset$.
2. Ponavljaj dok A ne povezuje graf G :
   (a) Nađi brid e koji povezuje dva vrha iz različitih komponenti povezanosti i dodaj ga u A. Pritom koristimo algoritam pretraživanja koji vraća rješenje u $O(n^2/t)$ iteracija.
3. Vrati A, skup bridova.

Teorem 5 *Neka je G neusmjeren graf u vektorskom modelu, sljedeći algoritam vraća MST grafa G u najviše $O(n)$ iteracija, ako je G povezan graf, inače algoritam ne staje.*

Algoritam za vektorski model:

1. Konstruiramo set bridova A na sljedeći način:
   (a) Inicijaliziramo $A = \emptyset$.
   (b) Postavimo $S = V$ (skup vrhova koji još nisu u nekoj komponenti povezanosti).
   (c) Postavimo $k = 0$ (broj već konstruiranih komponenti).
   (d) Ponavljaj dok $S \neq \emptyset$:
      i. Uzmi vrh $v \in S$ najvećeg stupnja i postavi $D = \{v\}$.
      ii. Prođi po svim susjedima od v, dodajući susjede ω u D i brid (v, ω) u A dok se ne dogodi jedno od sljedećeg:

A. Prošli smo po svim susjedima od v (došli smo do kraja liste susjeda), tada inkrementiramo k ($k = k + 1$), postavimo $C_k = D$ i $S = S \setminus D$.

B. Susjedni vrh ω je već dodjeljen nekoj komponenti C_j za $j \leq k$, tada dodamo D u C_j i izbacimo D iz S.

(e) Vraćamo k, A te komponente povezanosti $C_1, ..., C_k$.

2. Ponavljaj dok bridovi iz A ne razapinju graf G:

   (a) Odaberemo komponentu povezanosti C iz A najmanjeg stupnja (tako da minimizira $m_c = \sum_{i \in C} d_i$).

   (b) Nađi brid e iz C koji povezuje C s nekom drugom komponentom povezanosti iz A i dodaj ga u A. Koristimo algoritam za pretraživanje sa složenošću $O(\sqrt{m_C})$. Ako takav brid ne postoji algoritam ide u ∞.

3. Vrati skup bridova A.

5 Zaključak

U poglavlju 2 opisali smo standardne algoritme *Kruskal*ov i *Prim*ov kojima je najgora složenost $\mathcal{O}(n^2 log(n))$, odnosno $\mathcal{O}(n^2)$. Vidimo da *Boruvka* algoritam, opisan u poglavlju 4, ima bolju složenost u odnosu na prethodna dva algoritma. Također, kvantna verzija *Boruvka* algoritma daje nam poboljšanje u vidu složenosti koja je $\mathcal{O}(\sqrt{nm})$ u vektorskom modelu te $\mathcal{O}(n^{3/2})$ u matričnom modelu, gdje su nam n i m broj vrhova, odnosno broj bridova u grafu. Iz navedenog očito je poboljšanje u kvantnom algoritmu. U ovisnosti o broju bridova m proizlazi koja je implementacija pogodnija za koji graf, generalno zaključujemo da nam vektorski model daje manju složenost te kraće vrijeme izvršavanja.

Literatura

[1] Michael A. Nielsen, Isaac L. Chung. *Quantum Computation and Quantum Information* 2000.

[2] Christoph Durr, Mark Heiligman, Peter Hoyer, Mehdi Mhalla. *Quantum query complexity of some graph problems* 2004.
`https://arxiv.org/abs/quant-ph/0401091`

[3] Patrick J. Coles, Stephan Eidenbenz, Scott Pakin, Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, Nathan Lemons, Shizeng Lin, Andrey Lokhov, Alexander Malyzhenkov, David Mascarenas, Susan Mniszewski, Balu Nadiga, Dan O'Malley, Diane Oyen, Lakshman Prasad, Randy Roberts, Phil Romero, Nandakishore Santhi, Nikolai Sinitsyn, Pieter Swart, Marc Vuffray, Jim Wendelberger, Boram Yoon, Richard Zamora, and Wei Zhu *Quantum Algorithm Implementations for Beginners* 2018.