**Muhammad Al-Xorazmiy NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

**URGANCH FILIALI**

**Kurs ishi**

**Mavzu:**

**Onlayn oilaviy shajara algaritmini ishlab chiqish va dasturini loyihalash**

Talaba guruhi 942-20

F.I.O Raximov Shuxrat

Mundaraja

[Kirish 3](#_Toc75119)

[Maqsad 3](#_Toc75120)

[Mahsulot doirasi 3](#_Toc75121)

[Mahsulot istiqboli 3](#_Toc75126)

[Maxsus talablar 4](#_Toc75133)

[Tashqi interfeysga qo'yiladigan talablar 4](#_Toc75134)

[Foydalanuvchi interfeyslari 4](#_Toc75135)

[Uskuna interfeyslari 4](#_Toc75136)

[Dasturiy ta'minot interfeyslari 4](#_Toc75137)

[Aloqa interfeyslari 5](#_Toc75138)

[Funktsional talablar 5](#_Toc75139)

[UML 6](#_Toc75140)

Oilaviy shajara algaritmi blok sxema ko’rinishida  [7](#_Toc75141)

[Malumot kiritish yangi avlodlarni qo’shish va o’zgartirish 8](#_Toc75146)

[3.](#_Toc75147)

[3.](#_Toc75148)

[3.](#_Toc75149)

# Kirish

Bizning avlod deyarli barcha odamlar u yoki bu ma'lumotlarga ega bo'lgan davrda yashaydi. Shunday ekan, har birimiz hayotida axborot texnologiyalarining o‘rni katta. Ba’zan esa bu texnologiyalarsiz hayotimizni tasavvur etib bo‘lmaydi.

Bugungi kunda barcha zamonaviy qurilmalar yordamida hayotimiz ancha osonlashdi, eng muhimi - qulayroq! Axir, bir necha o'n yillar oldin, insoniyat hozir nimaga qodirligini orzu ham qila olmadi. Ishonamanki, IT ning paydo bo'lishi bilan har bir insonning hayoti ijobiy tomonga o'zgardi. Faqat atrof-muhitga kamroq zarar yetkazilsa, chunki barcha ma'lumotlar endi barcha turdagi elektron tashuvchilarda saqlanadi.

## Maqsad

## Ushbu hujjat loyiha menejerlariga, talablar bo'yicha muhandislarga, loyiha tahlilchilariga va mijozlar vakillariga mahallalalarda va boshqa davlat idoralarida onlayn oila shajarasini yaratish va boshqarish, manitoringini yuritish loyihasini yaratish jarayonida yordam berish uchun mo'ljallangan.

## Talablar va mumkin bo'lgan o'zgarishlar haqidagi barcha ma'lumotlar ushbu hujjatda saqlanadi va aniqlangan.

## Mahsulot doirasi

"Mahallarda, davlat idoralarida oilalar haqida malumot olish va shaxs avlodlari, ya’ni uning shajarasi haqida malumot kerak bo’lganda foydalanuvchi sifatida ma’lumot qidirayotgan paytda ishlatiladi.

Loyiha quyidagilarga qodir bo'lishi kerak:

● Ro'yxatdan o'tgan insonlarlar sonini aniqlash o'sish, kamayishini kuzatish

● Insonlar shajarasini kuzatish

●Oilaviy shajarani jadval ko’rinishida shakllantirish; ● tozalash jadvallarini saqlash va kuzatib borish

● Shajaralar haqida malumotlar saqlash.

## Mahsulot istiqboli

Bu loyiha davlat idoralarida axborot tizimining insonlarning oilaviy shajarasini boshqarishning yangi dasturiy mahsulotdir.

Tizim viloyatlarning davlat idoralarida , mahallalarda, bolalar bo'gchasida xizmat ko'rsatishi mumkin. Mahsulotning dastlabki qamrovi mahallalarda insonlarni ro'yxatga olish va ularning shajarasini kuzatib borish uchun barcha tegishli ma'lumotlar mavjud.

#### 

# Tashqi interfeysga qo'yiladigan talablar

### Foydalanuvchi interfeyslari

### TALAB-1: Bosh sahifa tizimga kirish opsiyasi bilan boshlanishi kerak.

### TALAB-2: Tizim foydalanuvchini avtorizatsiya qilishi va foydalanuvchini modulning asosiy sahifasiga yo'naltirishi kerak.

### TALAB-3: Asosiy sahifada foydalanuvchi tanlashi mumkin bo'lgan xususiyatlar bilan piktogramma bo'lishi kerak.

### TALAB-4: O'zgartirilgan ma'lumotlarni saqlash uchun tasdiqlash dialog oynasi paydo bo'lishi kerak.

### TALAB-5: Kiritilgan ma'lumotlarni saqlash uchun tasdiqlash dialog oynasi paydo bo'lishi kerak.

### TALAB-6: Tasdiqlash dialog oynasida ikkita variant bo'lishi kerak: Qo'llash va Bekor qilish.

### TALAB-7: Saqlanmagan ma'lumotlarni yo'qotishning oldini olish uchun ogohlantirish xabari paydo bo'lishi kerak.

### TALAB-8: Ogohlantirish xabarida ikkita variant bo'lishi kerak: Saqlash va E'tibor bermaslik.

### TALAB-9: Sahifalar o'xshash mavzuga mos kelishi kerak.

### Uskuna interfeyslari

### TALAB-1: Foydalanuvchilar tizimga kirish uchun Internetga ulangan kompyuterdan foydalanishlari kerak.

### TALAB-2: Funktsional standart klaviatura va sichqonchadan foydalanish kerak.

### TALAB-3: Bolalar bog'chasi boshqarish moduli veb-ga asoslangan dasturiy mahsulot bo'lib, apparat interfeyslariga ega emas. Serverga apparat ulanishi kompyuter va veb-serverdagi operatsion tizim tomonidan ta'minlanadi.

### Dasturiy ta'minot interfeyslari

### TALAB-1: Turli veb-brauzerlarni ishga tushirishga qodir bo'lgan har qanday operatsion tizimdan foydalanish mumkin.

### TALAB-2: Tizimni ishga tushirish uchun veb-brauzerlarning eng so'nggi versiyalaridan foydalanish kerak.

### TALAB-3: Ma'lumotlar ruxsatsiz kirishdan himoyalangan bo'lishi kerak.

### Aloqa interfeyslari

## TALAB-1: TCP/IP va HTTPS protokollari foydalanuvchi va veb-xosting xizmati o'rtasidagi aloqalarni osonlashtirish uchun ishlatilishi kerak.

## Funktsional talablar

#### Shaxslar hayotini tekshirish, tug'ilish va o’lishni ro'yxatga olish

#### Tavsif: Shaxslar hayotini tekshirish, tug'ilish va o’lishni ro'yxatga olish Ustuvorlik:

#### Kerak

#### Kuzatilish: qanoatlantiradi:

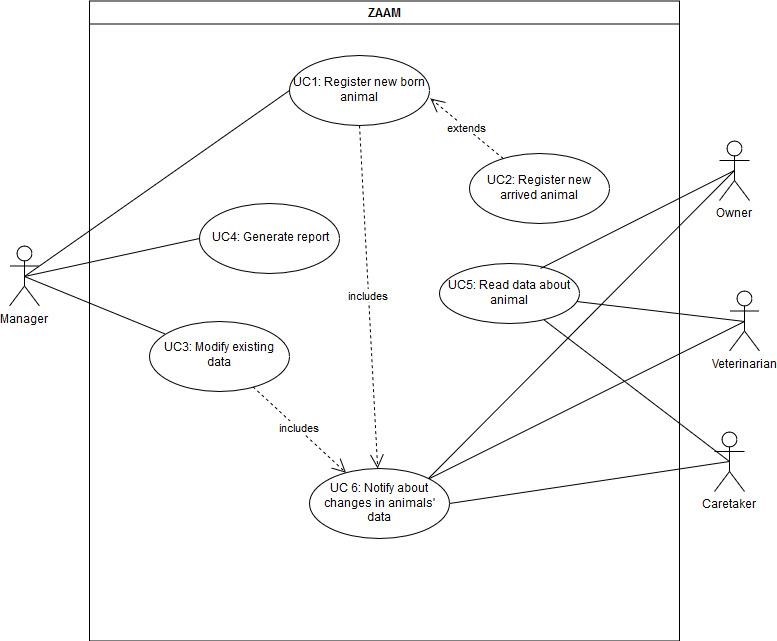
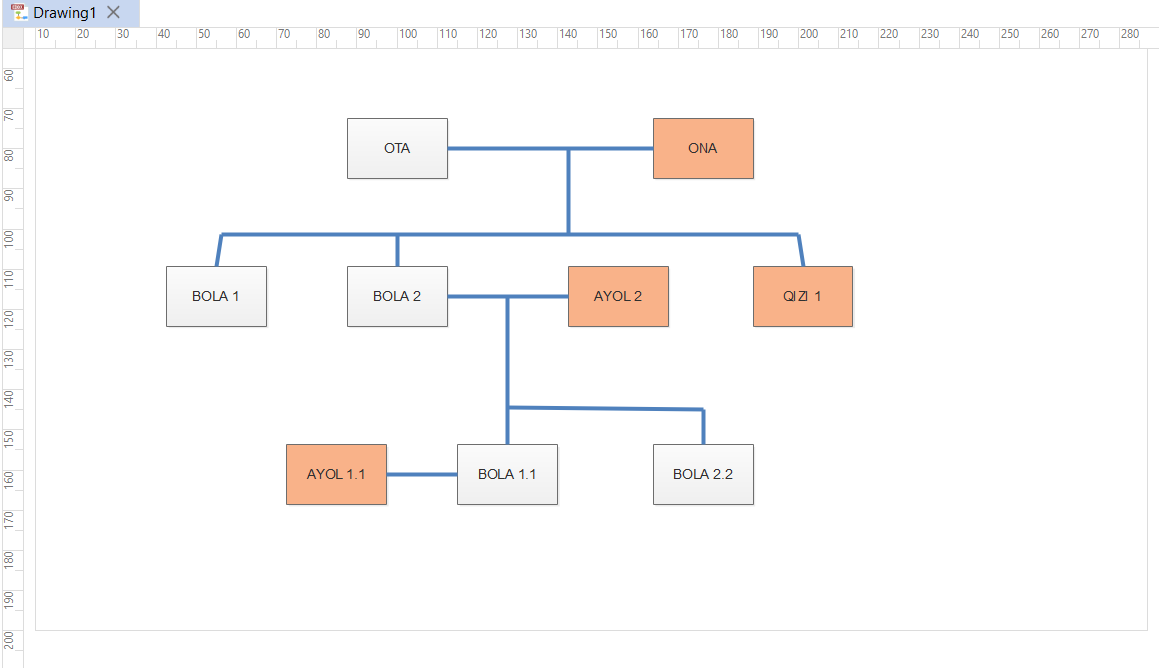
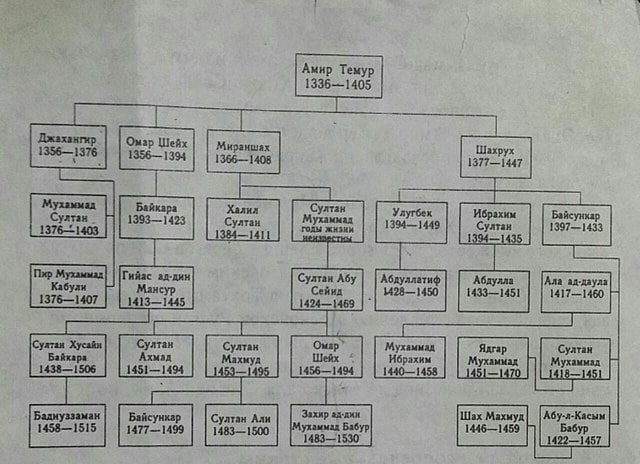


Figure 7: Bolalar tug’ilishini ro’yxatga olinishi uchun diagramma

UML loyiha diagrammasi



**Misol uchun Amir Temur shajarasini keltirsak, shu shajara orqali biz oila shajarasi algaritmini shakllantiramiz**

****

Oilaviy shajara algaritmi

Daraxtlarni chizish algoritmi

Men yaqinda elementlarning ierarxiyasini olib, ularni chiroyli daraxt tuzilishida chizmoqchi edim. Masalan, oila daraxti.

O'shanda men "Bu oson bo'ladi, men har bir tugunning X, Y o'rnini aniqlash uchun algoritm uchun Google-da qidiraman, keyin har bir tugunni ekranda chizish uchun biror narsa qilaman" deb o'yladim.

Ammo chiroyli daraxtlarni chizish uchun oddiy va oson bajariladigan algoritm uchun Googling juda qiyin.

Mening Google-dagi dastlabki urinishlarim meni **Reingold-Tilford** algoritmiga olib boradi.

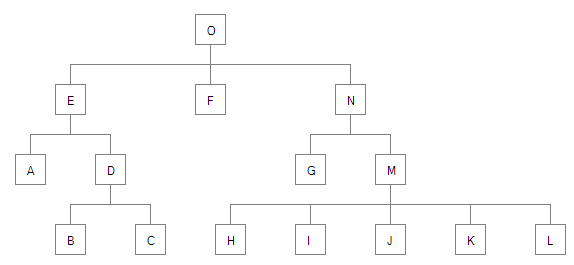
Bu men xohlagan narsaga o'xshardi, lekin men algoritmni tushunishda juda ko'p muammolarga duch keldim, shuning uchun uni o'zim tanlagan tilga (Python) aylantira oldim. Men topgan har bir maqola o‘zimdan ancha aqlliroq odam uchun yozilgandek tuyuldi va men ularning qilayotgan ishlarini kuzatishda va mantiqni Python ga aylantirishda biroz qiynaldim.

Tadqiqot hujjatlarini ko'rib chiqish va namuna kodini ko'rib chiqish uchun ko'p vaqt sarflaganimdan so'ng, men buni nihoyat tushundim deb o'ylayman. Shunday qilib, men oddiy atamalar yordamida daraxtlarni estetik jihatdan yoqimli tarzda chizish algoritmini tushuntirishga harakat qilaman.

Misol

Men doktor **Dobbsning** 1991 yilgi jurnal maqolasidagi misol daraxtidan foydalanmoqchiman. Men ushbu maqolani tushunish va o'z yo'limda ishlash uchun eng oson deb topdim va agar siz o'zingiz uchun chiroyli daraxtlar chizmoqchi bo'lsangiz, uni o'qib chiqishga arziydi. Shuningdek, u "estetik jihatdan yoqimli daraxt" deb hisoblangan narsalar haqida batafsilroq ma'lumotni o'z ichiga oladi.

Men tuzadigan shajara algaritmning namuna daraxti quyidagicha ko'rinadi:



Algoritmga kirish

Har bir tugunning Y pozitsiyasini aniqlash oson, shuning uchun men bu erda bu haqda batafsil ma'lumot bermayman. Asosiy muammo har bir tugun uchun mos X pozitsiyasini aniqlashdir.

Bu erda men har bir tugunning tegishli X pozitsiyasini aniqlash uchun foydalangan mantiqning juda qisqacha sharhi. Ehtimol, buni qabul qilib, X, Y qiymatlarini o'zingiz xohlagan tilda qanday olish kerakligini aniqlay olasiz, yoki men har bir qadamni keyinroq batafsilroq ko'rib chiqaman.

Daraxtni buyurtmadan keyin aylantiring

Har bir tugunga X qiymatini 0 belgilang, agar u chap tomondagi tugun boʻlsa, chap tomondagi tugun boʻlsa, leftSibling.X + 1 boʻlsa.

Har bir ota-ona tugun uchun biz tugunning bolalar ustida joylashganligini xohlaymiz. Bu birinchi bolaning X pozitsiyasi va oxirgi bolaning X pozitsiyasi o'rtasidagi o'rta nuqta bo'ladi.

Agar ota-onaning chap qardoshi bo'lmasa, uning X qiymatini ushbu o'rta nuqtaga o'zgartiring. Agar uning chap akasi bo'lsa, biz uni boshqa tugun xususiyatiga saqlaymiz. Men bu xususiyatni Mod deb atayman, chunki men boshqa misollarda shunday nomlanganini ko'raman.

Mod xususiyati bolalarning X qiymatlarini ota-ona tugunlari ostida markazlashtirish uchun qancha o'zgartirish kerakligini aniqlash uchun ishlatiladi va biz har bir tugunning yakuniy X qiymatini aniqlash uchun barcha hisob-kitoblarimizni bajarganimizda foydalaniladi. Bolalarni almashtirish uchun to'g'ri miqdorni aniqlash uchun aslida Parent.X - MiddleOfChildrenX ga o'rnatilishi kerak.

Ushbu daraxt avvalgi birodar daraxtlarining birortasiga zid emasligini tekshiring va agar kerak bo'lsa Mod xususiyatini sozlang. Bu joriy tugundagi har bir Y darajasida aylanishni anglatadi va tugunning chap tomonidagi birodarning eng o'ngdagi X qiymati joriy tugundagi har qanday bolaning eng chapdagi X qiymatini kesib o'tmasligini tekshiradi.

Ekrandan tashqarida bolalar chizilmasligini aniqlash uchun daraxt bo'ylab ikkinchi marta sayr qiling va agar kerak bo'lsa Mod xususiyatini sozlang. Mod xususiyati salbiy bo'lsa, bu sodir bo'lishi mumkin.

Har bir tugun uchun yakuniy X qiymatlarini aniqlash uchun daraxt bo'ylab uchinchi marta yuring. Bu tugunning X qiymati, shuningdek, ushbu tugunning barcha asosiy tugunlarining barcha Mod qiymatlari yig'indisi bo'ladi.

Endi har bir bosqichni batafsil ko'rib chiqamiz.

Daraxtning birinchi sayohati

Biz birinchi yurish uchun daraxtning oldindan buyurtmasini bajaramiz, ya'ni daraxtni pastdan yuqoriga, chapdan o'ngga qarab chizamiz. Daraxtning dastlabki chizilgan rasmini ko'rdingizmi? Tugunlarda A dan O gacha bo'lgan harflar mavjud. Bu tartibdan keyingi o'tish tugunlarni qayta ishlash tartibi.

Daraxtning bu iteratsiyasini 3 bosqichga bo'lish mumkin:

leftSibling.X + 1 ning har bir tuguniga X qiymatini belgilang, agar leftSibling bo‘lmasa 0

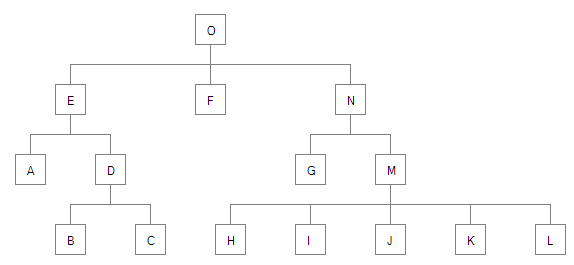
Bola tugunlarini ota-ona tugunlari ostiga o'tkazing

Bolalarning boshqa daraxtlar bilan to'qnashmasligiga ishonch hosil qiling

Dastlabki X qiymatlarini belgilash

Birinchi qadam, har bir tugun uchun "mahalliy X" qiymatini hisoblashdir. Bu faqat bolalar to'plamidagi X qiymati.

Shunday qilib, ushbu bosqichdan keyin har bir tugunning X qiymati quyidagicha ko'rinadi:



Buning uchun men foydalanadigan kod quyidagicha

private void CalculateInitialX(TreeNode node)

{

foreach (var child in node.Children)

CalculateInitialX(child);

// if there is a previous sibling in this set,

// set X to prevous sibling + designated distance

if (!node.IsLeftMost())

node.X = node.GetPreviousSibling().X + 1;

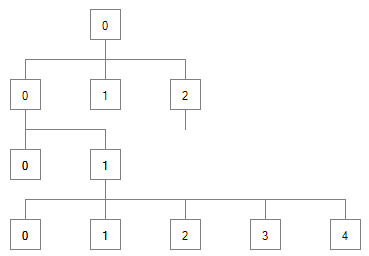
else

// if this is the first node in a set, set X to 0

node.X = 0;

}

Albatta, agar biz ushbu X qiymatlarida barcha tugunlarni chizgan bo'lsak, ular bir-biriga yopishib qoladi va siz daraxt ierarxiyasini ko'ra olmaysiz.



Shunday qilib, biz keyingi bosqichda bola tugunlarini ota-onasi ostida joylashtiramiz.

Mod mulki

Davom etishdan oldin, men "Mod" xususiyatini tushuntirishim kerak.

Ushbu qadamlarning qolgan qismi tugunni va uning barcha bolalarini, ba'zan bir necha marta almashtirishni o'z ichiga olishi mumkin. Buni amalga oshirish uchun biz odatda ushbu pastki daraxtdagi barcha tugunlarni aylanib chiqishimiz va ularning X qiymatlarini oshirishimiz kerak, biroq bu ishlash uchun yomon bo'lishi mumkin, ayniqsa kattaroq daraxtlar uchun.

Bunga yo'l qo'ymaslik uchun biz tugunga barcha bolalarini qancha masofaga siljitish kerakligini aytib berish uchun Mod xususiyatidan foydalanamiz, so'ngra buning oxirida daraxt bo'ylab ikkinchi marta sayr qilamiz va yakuniy X holatini aniqlaymiz. tugun.

Ushbu iteratsiya davomida tugunni va uning barcha bolalarini almashtirish uchun uning X va Mod xususiyatlarini xohlagancha oshiring.

Ota-onalar ostidagi bola tugunlarini joylashtirish

Keyingi qadam, asosiy tugun ostidagi asosiy tugunlarni markazlashtirishdir. Bu aslida juda oson.

Birinchidan, tugunni uning bolalari ustiga markazlashtiradigan X qiymatini toping.

Agar 1 ta bola bo'lsa, kerakli X qiymati bolaning X qiymati bilan bir xil bo'ladi.

Agar bir nechta bola bo'lsa, birinchi bolaning X va oxirgi bolaning X ni oling va ikkalasi orasidagi o'rta nuqtani toping.

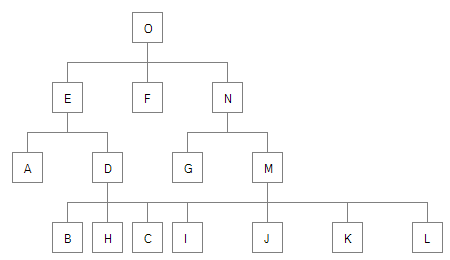
Keyin, bu ota-ona tugunining chap tomonida birodarlari borligini tekshiring. Men buni tugunning ota-onasini tekshirib, bu tugun ota-onaning farzandlaridagi birinchi tugunga teng yoki yo'qligini tekshirib ko'raman.

Agar bu ota-onaning bolalaridagi birinchi tugun bo'lsa, ota-onani bolalar ustidan markazlashtirish uchun X qiymatini kerakli X qiymatiga tenglashtiring.

Agar yo'q bo'lsa, bolalarni ota-onaga o'tkazish uchun tugunning Mod qiymatini node.X - desiredX ga belgilang.

Tugunlarning yakuniy X qiymatlarini hisoblash uchun uchinchi daraxt o'tish amalga oshirilgandan so'ng, bu bolalarni ota-ona tugunlari ostida to'g'ri joylashtiradi.

Biroq, bu daraxtlarning bir-birining ustiga chiqishiga to'sqinlik qilmaydi. Endi daraxtimizdagi B, C, H va I tugunlarini ko'rasizmi?



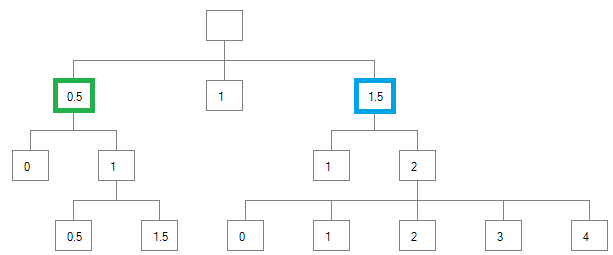
Tugun mojarolari tekshirilmoqda

Siz bu erga kelganingizdan so'ng, bu qadam ko'rinadigan darajada yomon emas.

Asosiy mantiq shundan iboratki, agar tugunning bolalari bo'lsa, biz barcha darajadagi bolalarni aylanib o'tishimiz va ushbu bolaga tayinlangan X qiymatlarining hech biri bir xil darajadagi ilgari joylashtirilgan bolalarga tayinlangan X qiymatlariga zid bo'lmasligiga ishonch hosil qilishimiz kerak.

Buni amalga oshirish uchun biz joriy tugunning barcha bolalarini aylantiramiz va har bir Y qiymati uchun minimal X pozitsiyasini yozamiz. Keyin ushbu tugunning chap tomonidagi har bir aka-uka uchun biz uning barcha bolalarini aylanib chiqamiz va har bir Y nuqtasida maksimal X holatini qayd qilamiz. Google menga buning uchun ajoyib atama borligini aytdi: daraxt konturi.

Shunday qilib, N (Moviy) tugun qayta ishlanayotgan vaqtda uning Chap konturi {1,5, 1, 0,0}, E tugunining o'ng konturi (yashil) esa {0,5, 1,0, 1,5} ga teng.



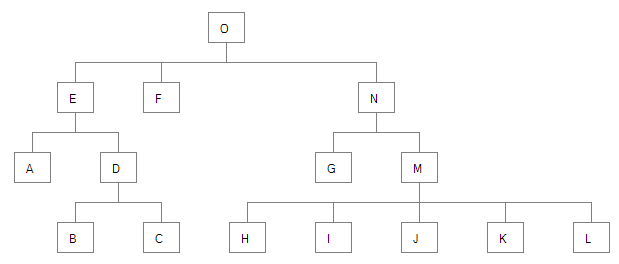
Yodda tutingki, ushbu tekshirish tugunning hozirgacha hisoblab chiqilganidek, biz dastlab tayinlagan mahalliy X emas, yakuniy X dan foydalanadi.

Biz birinchi va ikkinchi darajalarning bir-biriga mos kelishini ko'rishimiz mumkin, shuning uchun biz N tugunini chap konturi E tugunining o'ng konturiga zid bo'lmasligi uchun etarlicha uzoqqa siljitishimiz kerak.

Bu holda, ikkita kontur orasidagi eng katta bir-biriga mos keladigan masofa 1,5 ni tashkil qiladi va biz tugunlar bir-birining ustiga chizilmaganligiga ishonch hosil qilish uchun qo'shimcha 1 qo'shmoqchimiz, shuning uchun N tugunni 2,5 joyga siljitamiz.

Mojaroni hal qilgandan keyin o'rta daraxtlarni tuzatish

Oldingi qadam bir-birining ustiga chiqadigan tugunlarga ega bo'lish muammosini hal qildi, ammo yakuniy natija daraxtning o'rtasida juda ko'p bo'sh joylarni qoldiradi. Endi daraxtdagi F tugunining holatiga e'tibor bering:



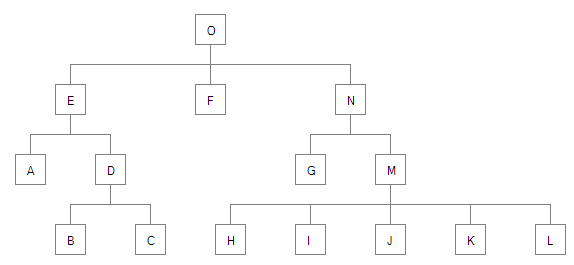
Biz N tugunni E bilan qarama-qarshi bo'lganligi sababli o'zgartirdik va endi biz bu masofani olib, ikkita qarama-qarshi tugunlar orasidagi barcha birodar tugunlari bo'ylab teng ravishda qo'llashimiz kerak.

Buning uchun biz bir-biriga qarama-qarshi bo'lgan ikkita tugunlar orasidagi masofani + 1 ga bo'lamiz va har bir o'rta tugunni shu qiymatga o'tkazamiz.

Bizning holatda biz tugunni 2,5 ga siljitdik va E va N tugunlari o'rtasida faqat 1 tugun (F) mavjud, shuning uchun biz F ni 2,5 / (1 + 1) ga o'tkazishimiz kerak, bu esa 1,25 ga olib keladi.

Va keyin o'rta tugunlarni siljitish orqali nizolarni keltirib chiqarmaganimizga ishonch hosil qilish uchun biz yana bir bor nizolarni tekshirishimiz kerak.

Natijada, daraxtlar zarur bo'lganidan biroz kattaroq bo'ladi, ammo ular hech qanday qarama-qarshi tugunlar yoki notekis miqdordagi bo'shliqlarsiz yaxshi hosil bo'ladi.

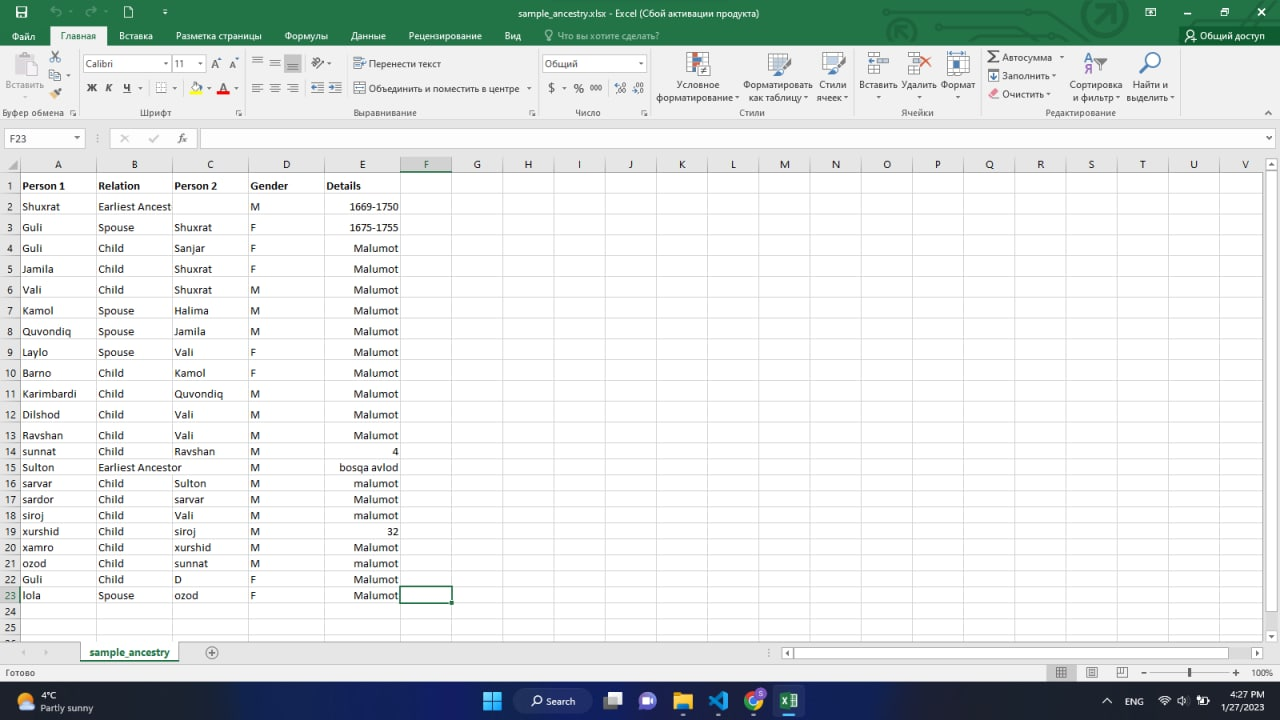


Bu algaritm orqali men o’zimning oila shajara algaritmimni python kodlar asosida shakllantirdim.

|  |
| --- |
| from graphviz import Digraph |
|  | import pandas as pd |
|  |  |
|  | ancestry = pd.read\_csv('sample\_ancestry.csv') |
|  | earl\_ans = ancestry.loc[ancestry['Relation'] == 'Earliest Ancestor', 'Person 1'].iloc[0] |
|  | ancestry['recorded\_ind'] = 0 # Flag for indicating individuals whose data has been recorded in the tree |
|  |  |
|  | incomp = [earl\_ans] |
|  | comp = [] |
|  |  |
|  | dot = Digraph(comment = 'Ancestry', graph\_attr = {'splines':'ortho'}) |
|  | node\_nm = [] |
|  |  |
|  | # Initializing first node |
|  | det = str(ancestry.loc[ancestry['Person 1'] == earl\_ans, 'Details'][0]) |
|  | g = ancestry.loc[ancestry['Person 1'] == earl\_ans, 'Gender'][0] |
|  | sh = 'rect' if g == 'M' else 'ellipse' |
|  | dot.node(earl\_ans, earl\_ans, tooltip = det, shape = sh) |
|  | node\_nm.append(earl\_ans) |
|  |  |
|  | ancestry.loc[ancestry['Person 1'] == earl\_ans, 'recorded\_ind'] = 1 |
|  |  |
|  | # max\_iter should be greater than number of generations |
|  | max\_iter = 5 |
|  |  |
|  | for i in range(0, max\_iter): |
|  | print(i) |
|  | temp = ancestry[ancestry['recorded\_ind'] == 0] |
|  |  |
|  | if len(temp) == 0: # Break loop when all individuals have been recorded |
|  | break |
|  | else: |
|  | temp['this\_gen\_ind'] = temp.apply(lambda x: 1 if x['Person 2'] in incomp else 0, axis = 1) |
|  |  |
|  | # Spouse Relationship |
|  | this\_gen = temp[(temp['this\_gen\_ind'] == 1) & (temp['Relation'] == 'Spouse')] |
|  | if len(this\_gen) > 0: |
|  | for j in range(0, len(this\_gen)): |
|  | per1 = this\_gen['Person 1'].iloc[j] |
|  | per2 = this\_gen['Person 2'].iloc[j] |
|  | det = str(this\_gen['Details'].iloc[j]) |
|  | g = this\_gen['Gender'].iloc[j] |
|  | sh = 'rect' if g == 'M' else 'ellipse' |
|  | with dot.subgraph() as subs: |
|  | subs.attr(rank = 'same') |
|  | subs.node(per1, per1, tooltip = det, shape = sh, fillcolor = "red") |
|  | node\_nm.append(per1) |
|  | subs.edge(per2, per1, arrowhead = 'none', color = "black:invis:black") |
|  |  |
|  | # Child Relationship |
|  | this\_gen = temp[(temp['this\_gen\_ind'] == 1) & (temp['Relation'] == 'Child')] |
|  | if len(this\_gen) > 0: |
|  | for j in range(0, len(this\_gen)): |
|  | per1 = this\_gen['Person 1'].iloc[j] |
|  | per2 = this\_gen['Person 2'].iloc[j] |
|  | det = str(this\_gen['Details'].iloc[j]) |
|  | g = this\_gen['Gender'].iloc[j] |
|  | sh = 'rect' if g == 'M' else 'ellipse' |
|  | dot.node(per1, per1, tooltip = det, shape = sh) |
|  | node\_nm.append(per1) |
|  | dot.edge(per2, per1) |
|  |  |
|  | comp.extend(incomp) |
|  | incomp = list(temp.loc[temp['this\_gen\_ind'] == 1, 'Person 1']) |
|  | ancestry['recorded\_ind'] = temp.apply(lambda x: 1 if (x['Person 1'] in incomp) | (x['Person 1'] in comp) else 0, axis = 1) |
|  |  |
|  | dot.format = 'svg' |
|  | dot.render('sample\_ancestry.gv.svg', view = True) |

Bu kod asosida shaxsning shajarasini yaratamiz. Buning uchun biz excel faylda shaxsning shajarasini shakllantiramiz.

Malumot kiritish yangi avlodlarni qo’shish va o’zgartirish



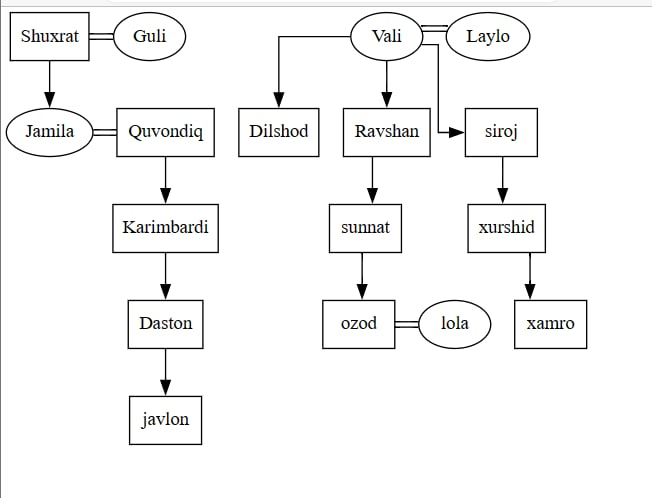
Bu yerda person 1 va person 2 lar avlodlar,gender person 1 ning jinsi haqida malumot, details person1 haqida qisqacha malumot(masalan tug’ilgan yili, yoshi, va boshqa malumotlar) beriladi. Jadvaldagi

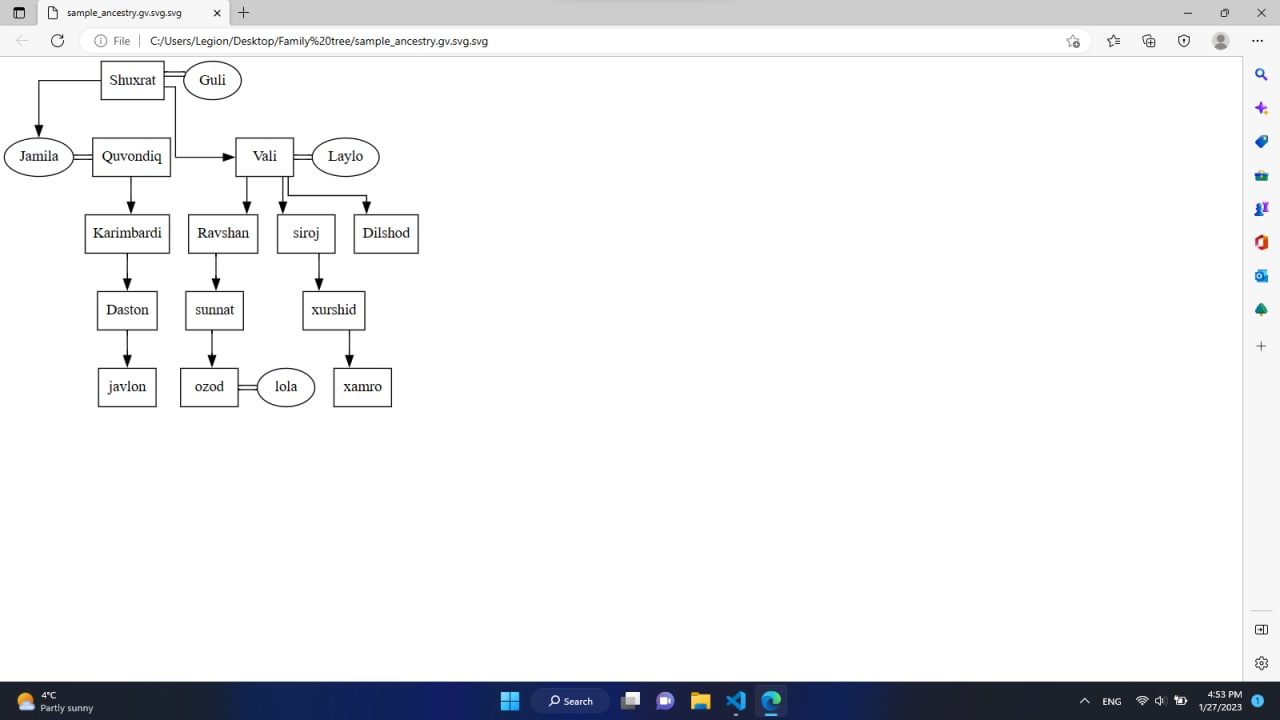
‘ lola Spouse ozod’ ozodning turmush o’rtog’i lola ;

‘ozod Child sunnat’ ozod sunnatning bolasi ;

‘sulton Earliest Ancestor ’ sulton qadimgi avlod degan ma’nolarni bildiradi

Keyin kodni ishga tushirsak bizga shaxsning shajarasini shakllantirib beradi.





Bu svg faylda shuini ko’rish mumkinki 3 inson o’z avlodlarini aytganda ular qarindosh yoki qarindosh emaslikini bilish mumkin(yoki teskarisi bir shaxsning avlodlari ko’rinishida ko’rishimiz mumkin).

**Bu oila algaritmi 2 ta xususiyatga ega:**

* Uni osongina tahrirlash kerak. Foydalanuvchilar ajdodlarning istalgan joyida har qanday shaxsning tafsilotlarini qo'shishlari, o'chirishlari yoki o'zgartirishlari kerak.
* Bu foydalanuvchilarga shaxs haqida qisqacha tavsif qo'shishga imkon berishi kerak. Masalan, shaxsning qaerda yashaganligi, uning ishi, asosiy hayotiy voqealar va boshqalar haqida ba'zi tafsilotlar.

Xulosa

Hozirda hayotni axborot texnologiyalarsiz tasavvur qilib bo’lmaydi. Foydalanuvchilarga qulayliklar yaratish uchun Axborot texnologiyalaridan keng qo’llaniladi. Ma’lumotlar ko’p, ularni tez va osongina topishda axborot texnologiyasi juda katta ro’l o’ynaydi. Shu sababli ma’lumotlarni internet orqali topishda algaritmlarning o’rni katta.

*Loyihanini yaratish davomida quydagi bilim va ko’nikmalar o’zlashtirildi:*

1. Algaritmlashni blok sxemada va kodda tasvirlash va; fayllar bilan ishlash borasida o’z ko’nikmamni oshirdim ;

2.Loyihani qanday tuzish va loyiha talablari qanday bo’lishini bilib oldim.