**ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ**

ՏՀՏԷ ինստիտուտի ԱԼ և Ծ ամբիոն

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ՏՏ924-1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ակադեմիական Խումբ

**ԿՈՒՐՍԱՅԻՆ ԱՇԽԱՏԱՆՔԻ ԱՌԱՋԱԴՐԱՆՔ**

Դիսկրետ մաթեմատիկա ալգորիթմներ և ծրագրավորում

Խորեն Կարենի Գալստյան

1. Թեմա ` Բուլյան ֆունկցիաներ: Կոմբիանտորիկա: Գրաֆների տեսություն:

2. Աշխատանքի նախանական տվյալները`\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Հաշվեբացատրագրի բովանդակությունը `\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ուսանող՝ \_\_\_\_\_\_\_\_\_\_\_ Խորեն Կարենի Գալստյան /Ստորագրություն/

անուն հայրանուն ազգանուն

Ղեկավար՝ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ստորագրություն/

անուն հայրանուն ազգանուն

Ամբիոնի վարիչ՝ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ստորագրություն/

անուն հայրանուն ազգանուն

Երևան – 2021

**ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ**

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

ՏՀՏԷ ինստիտուտի ԱԼ և Ծ ամբիոն

**ՀԱՇՎԵԲԱՑԱՏՐԱԳԻՐ**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ՏՏ924-1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ակադեմիական Խումբ

Դիսկրետ մաթեմատիկա ալգորիթմներ և ծրագրավորում *առարկայի կուրսային աշխատանք*

**1.Աշխատանքի թեման**՝

Բուլյան ֆունկցիաներ: Կոմբիանտորիկա: Գրաֆների տեսություն

**2.Աշխատանքի նախնական բովանդակությունը `\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Ուսանող՝ / / \_\_Խորեն Գալստյան Կարենի

/ստորագրություն/ /անուն, ազգանուն, հայրանուն/

Աշխատանքի ղեկավար՝ / / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ն. Ա. Սահակյան\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_

/ստորագրություն/ /անուն, ազգանուն, հայրանուն/

Ամբիոնի վարիչ ՝ / / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ս.Ս. Ավետիսյան\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

/ստորագրություն/ /անուն, ազգանուն, հայրանուն/

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Հաշվարկային և գրաֆիկական բաժինների կատարման որակ | Աշխատանքի կատարման համաչափությունը /0-10 միավոր/ | Աշխատանքի բովանդակության ներկայացման ուսանողի կարողությունը | Ուսանողի կողմից սեփական տեսակետը պաշտպանելու կարողությունը | Ուսանողի կողմից աշխատանքը ներկայացնելու հմտությունը և միջոցների կարողությունը | Ուսանողի խոսքի մշակույթը |
| 0-10 միավոր | 0-30 միավոր | 0-20 միավոր | 0-15 միավոր | 0-10 միավոր | 0-100 միավոր |
|  |  |  |  |  |  |

Հանձնաժողովի անդամներ` \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /ստորագրություն/

Հանձնաժողովի անդամներ` \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /ստորագրություն/

Երևան 2021

ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

[**ՆԵՐԱԾՈՒԹՅՈՒՆ** 6](#_Toc71539106)

[**ԳԼՈՒԽ 1** 8](#_Toc71539107)

[ՏԵՍԱԿԱՆ ՄԱՍ: ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ, ԿՈՄԲԻՆԱՏՈՐԻԿԱ ԵՎ ԳՐԱՖՆԵՐ 8](#_Toc71539108)

[**ԲԱԺԻՆ 1** 8](#_Toc71539109)

[ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ 8](#_Toc71539110)

[ԲՈՒԼՅԱՆ (ԿԱՄ 2-ԱԿԱՆ ) ՀԱՎԱՔԱԾՈՒ (ՎԵԿՏՈՐ) 8](#_Toc71539111)

[ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՅԻ ՍԱՀՄԱՆՈՒՄԸ 8](#_Toc71539112)

[ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐԻ ՆԵՐԿԱՅԱՑՄԱՆ ԱՂՅՈՒՍԱԿԱՅԻՆ ԵՂԱՆԱԿԸ 9](#_Toc71539113)

[ԷԱԿԱՆ և ԿԵՂԾ ՓՈՓՈԽԱԿԱՆՆԵՐ 9](#_Toc71539114)

[ԿԵՂԾ ՓՈՓՈԽԱԿԱՆՆԵՐԻ ՀԵՌԱՑՄԱՆ ԱԼԳՈՐԻԹՄ 9](#_Toc71539115)

[ՏԱՐՐԱԿԱՆ ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ 10](#_Toc71539116)

[ՏԱՐՐԱԿԱՆ ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐԻ ԲՆՈՒԹԱԳՐՈՒՄՆԵՐ 10](#_Toc71539117)

[ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՅԻ ԲԱՆԱՁԵՎԱՅԻՆ ՆԵՐԿԱՅԱՑՈՒՄ 11](#_Toc71539118)

[ԲԱՆԱՁԵՎԵՐ 11](#_Toc71539119)

[ԲԱՆԱՁԵՎԵՐԻ ՍԵՂՄ ՆԵՐԿԱՅԱՑՄԱՆ ԳԵՐԵԼԱՁԵՎԵՐ 12](#_Toc71539120)

[ԲԱՆԱՁԵՎԵՐԻՆ ՀԱՄԱՊԱՏԱՍԽԱՆՈՒՄ ԵՆ ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ 12](#_Toc71539121)

[ԲՈՒԼՅԱՆ ԳՈՐԾՈՂՈՒԹՅՈՒՆՆԵՐԻ ՆԱԽԱՊԱՏՎՈՒԹՅՈՒՆ 13](#_Toc71539122)

[ՀԱՄԱՐԺԵՔ ԲԱՆԱՁԵՎԵՐ 13](#_Toc71539123)

[ՏԱՐՐԱԿԱՆ ՀԱՄԱՐԺԵՔՈՒԹՅՈՒՆՆԵՐ 13](#_Toc71539124)

[ՀԻՄՆԱԿԱՆ ՀԱՄԱՐԺԵՔՈՒԹՅՈՒՆՆԵՐ 14](#_Toc71539125)

[ԴԻԶՅՈՒՆԿՏԻՎ ԵՎ ԿՈՆՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎԵՐ (ԴՆՁ ԵՎ ԿՆՁ) 15](#_Toc71539126)

[ԿՈՆՅՈՒՆԿՑԻԱ ԵՎ ԴԻԶՅՈՒՆԿՑԻԱ 15](#_Toc71539127)

[Դիցուք σk ∈{0, 1} և = , = , որտեղ ik∈ {1 , … , n}, k=1, 2, …, r (r ≥ 1 ,n ≥ 1): Այդ դեպքում K= ⋅ ⋅ … ⋅ բանաձևը կոչվում է կոնյունկցիա {x1, x2, …, xn} փոփոխականների բազմության վրա: 15](#_Toc71539128)

[ՏԱՐՐԱԿԱՆ ԿՈՆՅՈՒՆԿՑԻԱ ԵՎ ՏԱՐՐԱԿԱՆ ԴԻԶՅՈՒՆԿՑԻԱ 15](#_Toc71539129)

[ԴԻԶՅՈՒՆԿՏԻՎ ԵՎ ԿՈՆՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎԵՐ (ԴՆՁ ԵՎ ԿՆՁ) 15](#_Toc71539130)

[Եթե Ki –երը (I = 1 , … , s) զույգ առ զույգ իրարից տարբեր տարրական կոնյունկցիաներ են, ապա D = K1 ∨ K2 ∨ … ∨ Ks տեսքի արտահայտությունը կոչվում է դիզյունկտիվ նորմալ ձև (ԴՆՁ): 15](#_Toc71539131)

[ԴՆՁ-Ի ԵՎ ԿՆՁ-Ի ԿԱՌՈՒՑՈՒՄՆԵՐ ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ 15](#_Toc71539132)

[ԿՆՁ-ԻՑ ԱՆՑՈՒՄ ԴՆՁ-Ի ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ 16](#_Toc71539133)

[ԴՆՁ-ԻՑ ԱՆՑՈՒՄ ԿՆՁ-Ի ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ 16](#_Toc71539134)

[ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄ 16](#_Toc71539135)

[ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄԻ ԿԱՌՈՒՑՄԱՆ ԱՆՈՐՈՇ ԳՈՐԾԱԿԻՑՆԵՐԻ ՄԵԹՈԴԸ 16](#_Toc71539136)

[ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄԻ ԿԱՌՈՒՑՈՒՄԸ ԲԱՆԱՁԵՎԱՅԻՆ ՏԵՍՔԻ ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ 17](#_Toc71539137)

[ՖՈՒՆԿՑԻՈՆԱԼ ՏԱՐՐԵՐԻ ՍԽԵՄԱ 17](#_Toc71539138)

[ՖՈՒՆՑԻՈՆԱԼ ՍԽԵՄԱՅԻ ԲԱԶԻՍ 17](#_Toc71539139)

[ՖՈՒՆԿՑԻՈՆԱԼ ՍԽԵՄԱՆԵՐԻ ԵՐԿՐԱՉԱՓԱԿԱՆ ՆԵՐԿԱՅԱՑՈՒՄ 18](#_Toc71539140)

[**ԲԱԺԻՆ 2** 19](#_Toc71539141)

[ԿՈՄԲԻՆԱՏՈՐԻԿԱ 19](#_Toc71539142)

[ԿՈՄԲԻՆԱՏՈՐ ԲԱԶՄԱՊԱՏԿՄԱՆ ՍԿԶԲՈՒՆՔ 19](#_Toc71539143)

[ԿՈՄԲԻՆԱՏՈՐ ԳՈՒՄԱՐՄԱՆ ՍԿԶԲՈՒՆՔ 19](#_Toc71539144)

[ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ 20](#_Toc71539145)

[(n,k)-ԸՆՏՐՈՒԹՅՈՒՆ 20](#_Toc71539146)

[(n,k)-ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ 20](#_Toc71539147)

[ԿՐԿՆՈՒԹՅՈՒՆՆԵՐՈՎ ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ 20](#_Toc71539148)

[(n, k)-ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ ԿՐԿՆՈՒԹՅՈՒՆՆԵՐՈՎ 20](#_Toc71539149)

[ԶՈՒԳՈՐԴՈՒԹՅՈՒՆՆԵՐ 21](#_Toc71539150)

[ԶՈՒԳՈՐԴՈՒԹՅՈՒՆ ԿՐԿՆՈՒԹՅՈՒՆՆԵՐՈՎ 21](#_Toc71539151)

[ՏԱՐՐԱԿԱՆ ՆՈՒՅՆՈՒԹՅՈՒՆՆԵՐ 21](#_Toc71539152)

[ՆՅՈՒՏՈՆԻ ԵՐԿԱՆԴԱՄ 21](#_Toc71539153)

[ՊԱՍԿԱԼԻ ԵՌԱՆԿՅՈՒՆ 22](#_Toc71539154)

[**ԲԱԺԻՆ 3** 23](#_Toc71539155)

[ԳՐԱՖՆԵՐ 23](#_Toc71539156)

[ԳՐԱՖ 23](#_Toc71539157)

[ԳՐԱՖՆԵՐԻ ԳԱԳԱԹՆԵՐ ԵՎ ԿՈՂԵՐ 23](#_Toc71539158)

[ԳՐԱՖԻ ԵՐԿՐԱՉԱՓԱԿԱՆ ՄԵԿՆԱԲԱՆՈՒԹՅՈՒՆԸ 23](#_Toc71539159)

[ԿՈՂԻ ԾԱՅՐԱԿԵՏԵՐ, ԿՑՈՒԹՅՈՒՆ, ՀԱՐԵՎԱՆ ԳԱԳԹՆԵՐ ԵՎ ՀԱՐԵՎԱՆ ԿՈՂԵՐ 24](#_Toc71539160)

[ԳԱԳԱԹԻ ՀԱՐԵՎԱՆՈՒԹՅՈՒՆ 24](#_Toc71539161)

[ՀԱՆԳՈՒՅՑՆԵՐ ԵՎ ԶՈՒԳԱՀԵՌ ԿՈՂԵՐ 24](#_Toc71539162)

[ՊԱՐԶ ԿԱՄ ՍՈՎՈՐԱԿԱՆ ԳՐԱՖՆԵՐ 24](#_Toc71539163)

[ՄՈՒԼՏԻԳՐԱՖ 24](#_Toc71539164)

[ՊՍԵՎԴՈԳՐԱՖ 24](#_Toc71539165)

[ԿՈՂՄՆՈՐՈՇՎԱԾ ԳՐԱՖ 25](#_Toc71539166)

[ՀԻՊԵՐԳՐԱՖ 25](#_Toc71539167)

[ՆՇԱԳՐՎԱԾ ԿԱՄ ԿՇՌՎԱԾ ԳՐԱՖ 25](#_Toc71539168)

[ՀԱՐԵՎԱՆՈՒԹՅԱՆ ՄԱՏՐԻՑ 25](#_Toc71539169)

[ԿՑՈՒԹՅԱՆ ՄԱՏՐԻՑ 25](#_Toc71539170)

[ՀԱՐԵՎԱՆՈՒԹՅԱՆ ՑՈՒՑԱԿՆԵՐ 26](#_Toc71539171)

[ԳՐԱՖԻ ՆԵՐԿԱՅԱՑՄԱՆ ԵՂԱՆԱԿՆԵՐԸ 26](#_Toc71539172)

[ՀԱՎԱՍԱՐ ԳՐԱՖՆԵՐ 26](#_Toc71539173)

[ԻԶՈՄՈՐՖ ԳՐԱՖՆԵՐ 26](#_Toc71539174)

[ԵՆԹԱԳՐԱՖ, ԿՄԱԽՔԱՅԻՆ ԵՆԹԱԳՐԱՖ ԵՎ ԾՆՎԱԾ ԵՆԹԱԳՐԱՖ 26](#_Toc71539175)

[ԼՐԻՎ ԳՐԱՖ 27](#_Toc71539176)

[ԳԱԳԱԹԻ ԱՍՏԻՃԱՆ, ՄԵԿՈՒՍԱՑՎԱԾ ԵՎ ԿԱԽՎԱԾ ԳԱԳԱԹՆԵՐ 27](#_Toc71539177)

[ՆՎԱԶԱԳՈՒՅՆ ԵՎ ԱՌԱՎԵԼԱԳՈՒՅՆ ԱՍՏԻՃԱՆՆԵՐ 27](#_Toc71539178)

[ՀԱՄԱՍԵՌ ԳՐԱՖ 27](#_Toc71539179)

[ԵՐԿԿՈՂՄԱՆԻ ԳՐԱՖ 27](#_Toc71539180)

[ԳՐԱՖԱՅԻՆ ՀԱՋՈՐԴԱԿԱՆՈՒԹՅՈՒՆ 28](#_Toc71539181)

[ԱՆԿԱԽ ԳԱԳԱԹՆԵՐ, ԱՆԱԽՈՒԹՅԱՆ ԹԻՎ, ԱՆԿԱԽ ԿՈՂԵՐ 28](#_Toc71539182)

[ԶՈՒԳԱԿՑՈՒՄ, ԿԱՏԱՐՅԱԼ ԶՈՒԳԱԿՑՈՒՄ, 1-ՖԱԿՏՈՐ 28](#_Toc71539183)

[**ԳԼՈՒԽ 2** 30](#_Toc71539184)

[ԳՈՐԾՆԱԿԱՆ ՄԱՍ: ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ, ԿՈՄԲԻՆԱՏՈՐԻԿԱ ԵՎ ԳՐԱՖՆԵՐ 30](#_Toc71539185)

[**ԲԱԺԻՆ 1** 30](#_Toc71539186)

[ԻՍԿՈՒԹՅԱՆ ԱՂՅՈՒՍԱԿ 30](#_Toc71539187)

[ԻՍԿՈՒԹՅԱՆ ՎԵԿՏՈՐ 31](#_Toc71539188)

[ԷԱԿԱՆ ԵՎ ԿԵՂԾ ՓՈՓՈԽԱԿԱՆՆԵՐ 31](#_Toc71539189)

[ԿԱՏԱՐՅԱԼ ԴԻԶՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎ (ԿԴՆՁ) 32](#_Toc71539190)

[ԿԱՏԱՐՅԱԼ ԿՈՆՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎ (ԿԿՆՁ) 32](#_Toc71539191)

[ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄ 34](#_Toc71539192)

[ՖՈՒՆԿՑԻՈՆԱԼ ՏԱՐՐԵՐԻ ՍԽԵՄԱ (ՖՏՍ) 35](#_Toc71539193)

[**ԲԱԺԻՆ 2** 38](#_Toc71539194)

[ԽՆԴԻՐ 1 38](#_Toc71539195)

[ԽՆԴԻՐ 2 39](#_Toc71539196)

[**ԲԱԺԻՆ 3** 40](#_Toc71539197)

[**ԾՐԱԳԻՐ** 47](#_Toc71539199)

[**ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ** 73](#_Toc71539200)

# 

# 

# ՆԵՐԱԾՈՒԹՅՈՒՆ

Դիսկրետ մաթեմատիկան՝ որպես դիսկրետ օբյեկտների և դրանց հարաբերությունների բնագավառ, ընդգրկում է մի ամբողջ շարք ուղղություններ՝ թվերի տեսությունը, հանրահաշիվը, տրամաբանությունը, բուլյան հանրահաշիվը, ալգորիթմների տեսությունը, ավտոմատների տեսությունը, ֆունկցիոնալ համակարգերի տեսությունը, կոմբինատորիկան, գրաֆների և ցանցերի տեսությունը, կոդերի տեսությունը և այլն:

Կուրսային աշխատանքը պարունակում է բովանդակություն, և երկու գլուխներ:

Բովանդակությունում նշված են գլուխները, իրենց ենթագլուխներով:

Կուրսային աշխատանքի առաջին գլխում կուսումնասիրենք բուլյան ֆունկցիաները, կոմբինատորիկան և գրաֆները:

Կծանոթանանք բուլյան ֆունկցիաներին, բուլյան ֆունկցիաների ներկայացման աղյուսակային եղանակին, էական և կեղծ փոփոխականներին, տարրական բուլյան ֆունկցիաներին, բուլյան ֆունկցիաների բանաձևային ներկայացմանը, համարժեք բանաձևերին, դիզյունկտիվ և կոնյուկտիվ նորմալ ձևերին, ժեգալկինի բազմանդամին և ֆունկցիոնալ տարրերի սխեմային:

Կոմբինատորիկայի բաժնում կծանոթանանք կոմբինատոր գումարման և կոմբինատոր բազմապատկման սկզբունքներին, զուգորդություններին, ընտրություններին, տեղափոխություններին, Նյուտոնի երկանդամին, Պասկալի եռանկյունուն և այլն:

Գրաֆներ բաժնում կուսումնասիրենք հարևանության և կցության մատրիցները, կծանոթանանք գրաֆի ներկայացման եղանակներին:

Կուրսային աշխատանքի երկրորդ գլխում կկատարենք գործնական աշխատանք բուլյան ֆունկցիաների, կոմբինատորիկայի և գրաֆների վերաբերյալ: C++ լեզվով կգրենք ծրագրեր, որը կվերաբերի բուլյան ֆունկցիաներին և գրաֆներին:

# ԳԼՈՒԽ 1

# ՏԵՍԱԿԱՆ ՄԱՍ: ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ, ԿՈՄԲԻՆԱՏՈՐԻԿԱ ԵՎ ԳՐԱՖՆԵՐ

# ԲԱԺԻՆ 1

## ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ

Բուլյան ֆունկցիա (կամ տրամաբանական ֆունկցիա կամ հանրահաշվական տրամաբանության ֆունկցիա), n արգումենտից պատկերումը Bn → B, որտեղ B = {0, 1} –ը հանդիսանում է բուլյան բազմություն: Բուլյան բազմության {1, 0} տարրերը սովորաբար մեկնաբանվում են, որպես «ճիշտ» և «սխալ» տրամաբանական արժեքներով, չնայած ընդհանուր դեպքում նրանք դիտարկվում են որպես ֆորմալ սիմվոլներ, որոնք չեն կրում ինչ-որ առանձնակի իմաստ: Փոփոխականները, որոնք ընդունում են արժեքներ բուլյան բազմությունից, անվանում են բուլյան փոփոխականներ: Բուլյան ֆունկցիաները կոչվել են ի պատիվ մաթեմատիկոս Ջորջ Բուլի:

## ԲՈՒԼՅԱՆ (ԿԱՄ 2-ԱԿԱՆ ) ՀԱՎԱՔԱԾՈՒ (ՎԵԿՏՈՐ)

(α1, …, αn) հավաքածուն, որտեղ αI ∈ {0, 1}, i=1, 2, …, n, կոչվում է 2-ական կամ բուլյան հավաքածու (վեկտոր): Հավաքածուի տարրերը հաճախ կոչվում են բաղադրիչներ կամ կոորդինատներ: Սովորաբար, (α1, …, αn) բուլյան հավաքածուի բաղադրիչներն իրարից բաժանող ստորակետները բաց են թողնում:

## ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՅԻ ՍԱՀՄԱՆՈՒՄԸ

f(x1, x2, …, xn) ֆունկցիան, որը որոշված է Bn={0, 1}n բազմության վրա և արժեքներ է ընդունում {0, 1} բազմությունից, կոչվում է բուլյան հանրահաշվի ֆունկցիա կամ պարզապես բուլյան ֆունկցիա: P2n-ով նշանակվում է բոլոր բուլյան ֆունկցիաների բազմությունը, որոնք կախված են n փոփոխականներից, իսկ P2-ով՝ բոլոր բուլյան ֆունկցիաների բազմությունը:

## ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐԻ ՆԵՐԿԱՅԱՑՄԱՆ ԱՂՅՈՒՍԱԿԱՅԻՆ ԵՂԱՆԱԿԸ

Դիցուք ունենք f(, , …, ) բուլյան ֆունկցիա: Հաջորդաբար թվարկելով , , …, փոփոխականների արժեքների բոլոր հնարավոր , , …, թվային հավաքածուներն իրենց համարների աճման կարգով և յուրաքանչյուրի կողքին նշելով f(, , …, ) ֆունկցիայի համապատասխան f(, , …, ) արժեքը՝ կունենանք f(, , …, ) բուլյան ֆունկցիայի իսկության աղյուսակը:

## ԷԱԿԱՆ և ԿԵՂԾ ՓՈՓՈԽԱԿԱՆՆԵՐ

Բուլյան ֆունկցիայի ներկայացումը նույնիսկ իսկության վեկտորի միջոցով բավական հեռու է ռացիոնալ լինելուց, և խնդիր է առաջանում գտնել բուլյան ֆունկցիաների սեղմ ներկայացման ավելի արդյունավետ եղանակներ: Դրանցից մեկը բուլյան ֆունկցիայի այսպես կոչված կեղծ փոփոխականների հայտնաբերումն է և դրանց հեռացումը: Կարելի է ասել, որ f(, , …, ) ∈ P2n բուլյան ֆունկցիան էապես կախված է xi (1 ≤ i ≤ n) փոփոխականից, եթե գոյություն ունի , , … , , , … ,  փոփոխականների արժեքների այնպիսի ունի հավաքածու , , … , , , … , , որ f(, , … , , 0, , … , ) ≠ f(, , … , , 1, , … , ):

Այդ դեպքում xi-ն կոչվում է էական փոփոխական , հակառակ դեպքում ՝ կեղծ:

## ԿԵՂԾ ՓՈՓՈԽԱԿԱՆՆԵՐԻ ՀԵՌԱՑՄԱՆ ԱԼԳՈՐԻԹՄ

Կեղծ փոփոխականի սահմանումից հետևում է, որ դրանց առկայությունը կամ բացակայությունը որևէ ձևով չի ազդում ֆունկցիայի վրա և, հետևաբար, դրանք կարելի է անհրաժեշտության դեպքում հեռացնել կամ ավելացնել: Դիցուք հայտնի է, որ xi փոփոխականը f(x1, x2, …, xn) ֆունկցիայի կեղծ փոփոխական է: Դիտարկենք f ֆունկցիայի աղյուսակային տեսքը և նրանից կառուցենք նոր աղյուսակ՝ ջնջելով (α1 , … , αi-1 , 1 , αi+1 , … , αn) տեսքի բոլոր հնարավոր տողերը և ջնջելով xi արգումենտին համապատասխանող սյունը: Ստացված աղյուսակը կորոշի g(x1, …, xi-1, xi+1 , …, xn) ֆունկցիա: Ասում են, որ g ֆունկցիան ստացված է f ֆունկցիայից xi կեղծ փոփոխականի հեռացումով , իսկ f ֆունկցիան ստացված է g ֆունկցիայից xi կեղծ փոփոխականի ներմուծման ճանապարհով: Յուրաքանչյուր կեղծ փոփոխականի հեռացումով բուլյան ֆունկցիայի աղյուսակային ներկայացման ծավալը կրճատվում է կրկնակի անգամ:

## ՏԱՐՐԱԿԱՆ ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ

1 և 2-չաանի այն 11 բուլյան ֆունկցիաները, որոնք ներկայացված են հետևյալ երկու աղյուսակով, ընդունված է անվանել տարրական բուլյան ֆունկցիաներ:

|  |  |  |  |
| --- | --- | --- | --- |
| X | 0 |  | 1 |
| 0  1 | 0  0 | 1  0 | 1  1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x y | x ∨ y | xy | x ⊕ y | x ≡ y | x → y | x | y | x ↓ y |
| 0 0  0 1  1 0  1 1 | 0  1  1  1 | 0  0  0  1 | 0  1  1  0 | 1  0  0  1 | 1  1  0  1 | 1  1  1  0 | 1  0  0  0 |

## ՏԱՐՐԱԿԱՆ ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐԻ ԲՆՈՒԹԱԳՐՈՒՄՆԵՐ

* 0 և 1 ֆունկցիաները կոչվում են համապատասխանաբար նույնաբար 0 և նույնաբար 1 ֆունկցիաներ,
* x ֆունկցիան կոչվում է նույնական ֆունկցիա,
* ֆունկցիան կոչվում է x-ի ժխտում, նշանակվում է ¬x = = x’ նշաններով և կարդացվում է “ոչ x”,
* x ∨ y ֆունկցիան կոչվում է x-ի և y-ի դիզյունկցիա, նշանակվում է x ∨ y = x + y = max{x, y} և կարդացվում է “x կամ y”,
* xy ֆունկցիան կոչվում է x-ի և y-ի կոնյունկցիա, նշանակվում է x ∧ y = xy = min{x, y} և կարդացվում է “x և y”,
* x ⊕ y ֆունկցիան կոչվում է x-ի և y-ի մոդուլ 2-ով գումարում , նշանակվում է x ⊕ y = x + y և կարդացվում է “x գումարած y”,
* x ≡ y ֆունկցիան կոչվում է x-ի և y-ի համարժեքություն , նշանակվում է (x ≡ y) = (x ~ y) = (x ↔ y) և կարդացվում է “x-ը համարժեք է y-ին”,
* x → y ֆունկցիան կոչվում է x-ի և y-ի իմպլիկացիա, նշանակվում է (x → y) = (x ⊃ y) և կարդացվում է “x-ից հետևում է y-ը”,
* x | y ֆունկցիան կոչվում է x-ի և y-ի Շեֆերի շտրիխ, նշանակվում է x | y և կարդացվում է “ոչ x կամ ոչ y” (քանի որ x | y= ∨ ) կամ “x-ը չ y-ը համատեղելի չեն”: Տեխնիկական գրականության մեջ x | y ֆունկցիան սովորաբար կոչվում է հակակոնյունկցիա:
* x ↓ y ֆունկցիան կոչվում է x-ի և y-ի Պիրսի սլաք, նշանակվում է x ↓ y և կարդացվում է “ոչ x, ոչ y” (քանի որ x ↓ y = ∧ ), կամ “ոչ x և ոչ y”: Տեխնիկական գրականության մեջ x ↓ y սովորաբար կոչվում է հակադիզյունկցիա, ինչպես նաև Դագերի ֆունկցիա և Վեբի ֆունկցիա:
* Նույնաբար 0 և 1 ֆունկցիաները դիտարկվում են որպես 0 տեղանի ֆունկցիաներ , այսինքն՝ ֆունկցիաներ, որոնց փոփոխականները պատկանում են դատարկ բազմությանը:

## ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՅԻ ԲԱՆԱՁԵՎԱՅԻՆ ՆԵՐԿԱՅԱՑՈՒՄ

## ԲԱՆԱՁԵՎԵՐ

Բուլյան ֆունկցիայի ներկայացման առավել արդյունավետ ձևերից մեկը բանաձևային ներկայացումն է, որի հիմքում ընկած տարրական բուլյան ֆունկցիաներին համապատասխանող այսպես կոչված տարրական բանաձևերը: Տարրական բանաձևերից մակածման եղանակով ավելի բարդ բանաձևերի կառուցումը իրականացվում է հետևյալ կերպ.

Դիցուք F={¬, &, ∨, ⊕, ≡, →, |, ↓}, իսկ X-ը՝ փոփոխականների որևէ այբուբեն է:

1. Կամայական x ∈ X փոփոխական կոչվում է բանաձև F-ի վրա;
2. Եթե A-ն և B-ն բանաձևեր են F-ի վրա, ապա (¬A), (A & B), (A ∨ B), (A ⊕ B), (A ≡ B), (A → B), (A | B) և (A ↓ B) արտահայտությունները նույնպես բանաձևեր են F-ի վրա;
3. Այլ բանաձևեր չկան, և F համակարգում ամեն մի բանաձև ստացվում է առաջին երկու կանոնի միջոցով:

## ԲԱՆԱՁԵՎԵՐԻ ՍԵՂՄ ՆԵՐԿԱՅԱՑՄԱՆ ԳԵՐԵԼԱՁԵՎԵՐ

Բանաձևերի ավելի սեղմ ներկայացման նպատակով հաճախ ընդունում են հետևյալ պայմանավորվածությունները.

* Բանաձևերի արտաքին փակագծերը բաց են թողնվում;
* (¬ A) բանաձևը ներկայացվում է տեսքով;
* (A & B) բանաձևը ներկայացվում է (AB) տեսքով;
* Ենթադրվում է, որ ¬ առնչությունը ավելի ուժեղ է, քան ցանկացած երկտեղանի այլ առնչություն F-ի վրա;
* Ենթադրվում է, ¬ -ից հետո & առնչությունն ավելի ուժեղ է, քան ցանկացած այլ առնչություն F-ի վրա;

## ԲԱՆԱՁԵՎԵՐԻՆ ՀԱՄԱՊԱՏԱՍԽԱՆՈՒՄ ԵՆ ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ

Քանի որ բանաձևերի կառուցման մակածման հիմքում ընկած են տարրական բուլյան ֆունկցիաները, ապա դա իսկության աղյուսակների միջոցով կարելի է քայլ առ քայլ կառուցել տվյալ բանաձևին համապատասխան բուլյան ֆունկցիայի իսկության աղյուսակը: Այլ կերպ ասած, կամայական բանաձևի միարժեք ձևով համապատասխանում է բուլյան ֆունկցիա, ընդ որում, տարբեր բանաձևերի երբեմն կարող են համապատասխանել հավասար ֆունկցիաներ:

## ԲՈՒԼՅԱՆ ԳՈՐԾՈՂՈՒԹՅՈՒՆՆԵՐԻ ՆԱԽԱՊԱՏՎՈՒԹՅՈՒՆ

Բանաձևերը միարժեք վերականգնելու համար գոյություն ունի տարրական բուլյան ֆունկցիաների նախապատվություն (¬, ∧, ∨, →) հաջորդականությամբ, որը թույլ է տալիս որոշ փակագծեր բաց թողնել:

## ՀԱՄԱՐԺԵՔ ԲԱՆԱՁԵՎԵՐ

Երկու բանաձևեր կոչվում են համարժեք, եթե նրանք իրացնում են միևնույն բուլյան ֆունկցիան:

## ՏԱՐՐԱԿԱՆ ՀԱՄԱՐԺԵՔՈՒԹՅՈՒՆՆԵՐ

Կառուցելով համապատասխան աղյուսակները՝ կարելի է համոզվել, որ ճիշտ են հետևյալ համարժեքությունները.

1. Կոմուտատիվություն, որտեղ \* նշանը կարող է լինել &, ∨, ⊕, ≡, |, ↓ նշաններից կամայականը. x \* y = y \* x:
2. Ասոցիատիվություն, որտեղ \* նշանը կարող է լինել &, ∨, ⊕, ≡ նշաններից կամայականը. (x \* y) \* z = x \* (y \* z):
3. Կոնյունկցիայի դիստրիբուտիվություն դիզյունկցիայի նկատմամբ. x(y ∨ z) = xy ∨ xz:
4. Դիզյունկցիայի դիստրիբուտիվություն կոնյունկցիայի նկատմամբ. x ∨ yz = xy ∨ xz:
5. Կոնյունկցիայի դիստրիբուտիվություն մոդուլ 2-ով գումարման նկատմամբ. x(y ⊕ z) = xy ⊕ xz:
6. Կլանման կանոններ. x ∨ xy = x , x(x ∨ y) = x:
7. Կրճատման կանոններ. x ∨ y = x ∨ y , x( ∨ y) = xy:
8. 0, 1, x և ֆունկցիաների համարժեքությունները.



1. Դիզյունկցիայի, կոնյունկցիայի և դրանց ժխտումների համարժեքությունները (Մորգանի կանոններ).

x ∨ y = , x⋅y = ,

= ⋅ , = ∨ :

1. ⊕, ≡, →, | և ↓ ֆունկցիաների համարժեքությունները.

x ⊕ y = x ∨ y = (x ∨ y)(∨ ),

= (x ≡ y) = xy ∨ ⋅ = (x ∨ )( ∨ y),

(x ≡ y) = = xy ∨ ⋅ = (x ∨ )( ∨ y),

= x ∨ y = x ⊕ y = (x ∨ y)( ∨ ),

x → y = ∨ y = (x⋅y ⊕ x) ⊕ 1; = x,

x | y = = ∨ ; =xy,

x ↓ y = = ⋅ ; = x ∨ y:

## ՀԻՄՆԱԿԱՆ ՀԱՄԱՐԺԵՔՈՒԹՅՈՒՆՆԵՐ

x ∨ y = , = ⋅

x⋅y = , = ∨ :

x ⊕ y = x ∨ y = (x ∨ y)(∨ ) = ,

= (x ≡ y) = xy ∨ ⋅ = (x ∨ )( ∨ y) = (x ≡ y) ,

(x ≡ y) = xy ∨ ⋅ = (x ∨ )( ∨ y) = ,

= x ∨ y = x ⊕ y = (x ∨ y)( ∨ ) ,

x → y = ∨ y = (x⋅y ⊕ x) ⊕ 1; = x ,

x | y = = ∨ ; =xy ,

x ↓ y = = ⋅ ; = x ∨ y:

## ԴԻԶՅՈՒՆԿՏԻՎ ԵՎ ԿՈՆՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎԵՐ (ԴՆՁ ԵՎ ԿՆՁ)

## ԿՈՆՅՈՒՆԿՑԻԱ ԵՎ ԴԻԶՅՈՒՆԿՑԻԱ

## Դիցուք σk ∈{0, 1} և = , = , որտեղ ik∈ {1 , … , n}, k=1, 2, …, r (r ≥ 1 ,n ≥ 1): Այդ դեպքում K= ⋅ ⋅ … ⋅ բանաձևը կոչվում է կոնյունկցիա {x1, x2, …, xn} փոփոխականների բազմության վրա:

Համանմանորեն, D= ∨ ∨ … ∨ բանաձևը կոչվում է դիզյունկցիա:

## ՏԱՐՐԱԿԱՆ ԿՈՆՅՈՒՆԿՑԻԱ ԵՎ ՏԱՐՐԱԿԱՆ ԴԻԶՅՈՒՆԿՑԻԱ

Կոնյունկցիան և դիզյունկցիան կոչվում են տարրական, եթե կրկնվող փոփոխականներ չեն պարունակում:

## ԴԻԶՅՈՒՆԿՏԻՎ ԵՎ ԿՈՆՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎԵՐ (ԴՆՁ ԵՎ ԿՆՁ)

## Եթե Ki –երը (I = 1 , … , s) զույգ առ զույգ իրարից տարբեր տարրական կոնյունկցիաներ են, ապա D = K1 ∨ K2 ∨ … ∨ Ks տեսքի արտահայտությունը կոչվում է դիզյունկտիվ նորմալ ձև (ԴՆՁ):

Համանմանորեն, K = D1 ⋅ D2 ⋅ … ⋅ Ds տեսքի արտահայտությունը, որտեղ Di – երը (i = 1 , … , s) զույգ առ զույգ իրարից տարբեր տարրական դիզյունկցիաներ են, կոչվում է կոնյունկտիվ նորմալ ձև (ԿՆՁ):

## ԴՆՁ-Ի ԵՎ ԿՆՁ-Ի ԿԱՌՈՒՑՈՒՄՆԵՐ ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ

Եթե բուլյան ֆունկցիան տրված է բանաձևային տեսքով, ապա նրա դիզյունկտիվ և կոնյունկտիվ նորմալ ձևերը կարելի է կառուցել՝ օգտագործելով տարրական և հիմնական համարժեքությունները:

## ԿՆՁ-ԻՑ ԱՆՑՈՒՄ ԴՆՁ-Ի ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ

ԿՆՁ-ԻՑ ԴՆՁ-ին անցնելու համար անհրաժեշտ է օգտագործել դիստրիբուտիվության x⋅(y ∨ z) = x⋅y ∨ x⋅z օրենքը՝ համատեղելով այն x⋅x = x, x⋅ = 0, A⋅0 = 0, A ∨ 0 = A համարժեք ձևափոխություններով և կլանման A ∨ A⋅B = A կանոնով:

## ԴՆՁ-ԻՑ ԱՆՑՈՒՄ ԿՆՁ-Ի ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ

ԴՆՁ-ից ԿՆՁ-ին անցնելու համար անհրաժեշտ է օգտագործել դիստրիբուտիվության x ∨ y⋅z = (x ∨ y)⋅(x ∨ z) օրենքը՝ համատեղելով այն x ∨ x = x, x ∨ = 1, A ∨ 1 = 1, A⋅1 = A համարժեք ձևափոխություններով և կլանման A⋅(A ∨ B) = A կանոնով:

## ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄ

Ժեգալկինի բազմանդամ (կամ 2 մոդուլով բազմանդամ) կանվանենք K1 ⊕ K2 ⊕ … ⊕ Ks տեսքի արտահայտությունը, որտեղ Ki-երը (i = 1, 2, …, s) զույգ առ զույգ իրարից տարբեր մոնոտոն տարրական կոնյունկցիաներ են փոփոխականների x1, x2, …, xn բազմության վրա: Դիտարկվում է նաև Ժեգալկինի բազմանդամ s=0 դեպքի համար: Այդպիսի բազմանդամը նշանակում են 0 և համարում են , որ այն նույնն է , ինչ 0 հաստատունը:

## ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄԻ ԿԱՌՈՒՑՄԱՆ ԱՆՈՐՈՇ ԳՈՐԾԱԿԻՑՆԵՐԻ ՄԵԹՈԴԸ

Դիցուք P(x1, x2, …, xn)-ը f(x1, x2, …, xn) բուլյան ֆունկցիան իրացնող Ժեգալկինի որոնելի բազմանդամն է: Ընդհանուր տեսքով P(x1, x2, …, xn)-ը կարող ենք ներկայացնել հետևյալ կերպ՝

P(x1, x2, …, xn) = β0 ⋅K0 ⊕ β1⋅K1 ⊕ … ⊕ ⋅ ,

որտեղ Kj-ով նշանակված է j համար ունեցող մոնոտոն տարրական կոնյունկցիան (j =1, 2, … ,-1), իսկ β0,β1, - երը այն անորոշ գործակիցներն են, որոնք անհրաժեշտ է գտնել: x1, x2, …, xn փոփոխականների կամայական α1, α2, …, αn հավաքածուի համար կազմելով P(α1, α2, …, αn) = f(α1, α2, …, αn) հավասարումը, կստանանք 2n հավասարումների համակարգ 2n անհայտներով, որն ունի միակ լուծում: Լուծելով այս համակարգը՝ կգտնենք β0, β1, …, գործակիցները, և հետևաբար՝ P(x1 , x2, …, xn) բազմանդամը:

## ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄԻ ԿԱՌՈՒՑՈՒՄԸ ԲԱՆԱՁԵՎԱՅԻՆ ՏԵՍՔԻ ՀԱՄԱՐԺԵՔ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐՈՎ

Տրված բանաձևը, որն իրացնում է տրված բուլյան ֆունկցիան, համարժեք ձևափոխություններով բերենք այնպիսի տեսքի, որտեղ մասնակցում են միայն {&, ¬} գործողությունները: Այնուհետև, ստացված բանաձևում ամենուրեք տեսքի ենթաբանաձևերը փոխարինենք A ⊕ 1 տեսքի համարժեք բանաձևերով: Բացելով փակագծերը և օգտագործելով A⋅(B ⊕ C) = A⋅B ⊕ A⋅C դիստրիբուտիվության օրենքը, ինչպես նաև A⋅A = A, A⋅1 = A, A ⊕ A = 0, A ⊕ 0 = A համարժեքությունները՝ կստանանք տրված բանաձևին համարժեք Ժեգալկինի բազմանդամը:

## ՖՈՒՆԿՑԻՈՆԱԼ ՏԱՐՐԵՐԻ ՍԽԵՄԱ

Ֆունկցիոնալ տարրերի սխեման ցանց է, որի մուտքային գագաթներին վերագրված են փոփոխականների նիշեր (երբեմն նաև փոփոխականների ժխտումների նիշեր կամ հաստատուններ), իսկ մնացած գագաթներին վերագրված են ֆունկցիոնալ նշաններ այնպես, որ տվյալ գագաթը մտնող սլաքների թիվը հավասար լինի այդ գագաթին վերագրված ֆունկցիոնալ նշանի տեղերի քանակին (օրինակ ժխտումը մի տեղանի է, իսկ դիզյունկցիան՝ երկտեղանի):

## ՖՈՒՆՑԻՈՆԱԼ ՍԽԵՄԱՅԻ ԲԱԶԻՍ

Ֆունկցիոնալ նշանների բազմությունը, որոնցով նշված են սխեմայի գագաթները, կոչվում է ֆունկցիոնալ սխեմայի բազիս: Մասնավորապես, {∨, &, ¬} բազիսը կոչվում է ստանդարտ բազիս:

## ՖՈՒՆԿՑԻՈՆԱԼ ՍԽԵՄԱՆԵՐԻ ԵՐԿՐԱՉԱՓԱԿԱՆ ՆԵՐԿԱՅԱՑՈՒՄ

Ստորև բերված նկարներից առաջինում ֆունկցիոնալ սխեմայի մուտքային գագաթները նշված են լուսավոր շրջանակներով, ներքին գագաթները՝ կրկնակի շրջանակներով, իսկ ելքային գագաթը՝ սև շրջանակով: Երկրորդ նկարում բերված է ֆունկցիոնալ սխեմայի ներկայացման մեկ այլ եղանակ, որտեղ ներքին գագաթները փոխարինված են եռանկյուններով, որոնց ներսում գրված է համապատասխան ֆունկցիոնալ նշանը: Յուրաքանչյուր եռանկյան վերևի հորիզոնական կողի վրա միացված են ֆունցկիոնալ տարրի մուտքերը, իսկ այդ կողի հանդիպակաց գագաթից դուրս է բերված ելքը: Ֆունկցիոնալ սխեմայի ելքն առանձնացված է ֆունկցիոնալ տարրերից և, ինչպես նախորդ նկարում, նշված է սև շրջանակով:



# ԲԱԺԻՆ 2

## ԿՈՄԲԻՆԱՏՈՐԻԿԱ

Կոմբինատորիկա կամ կոմբինատորային անալիզ, մաթեմատիկայի բաժին է, որն ուսումնասիրում է օբյեկտների դիսկրետությունը, բազմությունների (էլեմենտների ընտրություններ, վերադասավորումներ, վարիացիաներ, տարբեր կոնֆիգուրացիաների վերահաշվարկներ) կառուցվածքների հատկությունները, դրանց հետ կապված քանակական մեծությունները: Կոմբինատորիկան կապված է մաթեմատիկայի այլ բաժինների հետ, ինչպիսիք են՝ հանրահաշիվը, երկրաչափությունը, հավանականությունների տեսությունը, և կիրառվում է գիտության տարբեր ոլորտներում:

«Կոմբինատորիկա» տերմինը գործածության մեջ է դրվել Գոթֆրիդ Լայբնիցի կոմից 1666 թվականին հրատարակած «Դատողություններ կոմբինատորային արվեստի մասին» աշխատությունում:

## ԿՈՄԲԻՆԱՏՈՐ ԲԱԶՄԱՊԱՏԿՄԱՆ ՍԿԶԲՈՒՆՔ

Դիցուք տված է պատահարների E1, E2, …, Em հաջորդականությունը, որտեղ E1 պատահարն ունի n1 հավանական ելքեր, և եթե E1, E2, …, Ek-1 պատահարներն իրականացվել են, ապա Ek պատահարն ունի nk հավանական ելքեր: Այդ դեպքում պատահարների ամբողջ հաջորդականությունն ունի n1 × n2 × … × nm հավանական ելքեր:

## ԿՈՄԲԻՆԱՏՈՐ ԳՈՒՄԱՐՄԱՆ ՍԿԶԲՈՒՆՔ

Դիցուք S-ը և T-ն բազմություններ են: Այն տարրերի քանակը, որոնք կարելի է ընտրել S-ից կամ T-ից, հավասար է + - : Բազմությունների լեզվով, = + - :

## ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ

Տեղափոխությունը որևէ բազմության տարրերի վերադասավորումներն են այլ հաջորդականություններով: Բոլոր հնարավոր տեղափոխությունների քանակը հաշվելու համար բավական է օգտվել կոմբինատոր բազմապատկման սկզբունքից:

## (n,k)-ԸՆՏՐՈՒԹՅՈՒՆ

(n,k)-ընտրությունը որևէ k տարրերի ընտրությունն է n հզորություն ունեցող S բազմությունից:

## (n,k)-ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ

­(n,k)-տեղափոխությունները (n, k)-ընտրություններ են և դրանցից յուրաքանչյուրում տարրերի վերադասավորումներ՝ այլ հաջորդականություններով:

P(n,k)-ով (կամ -ով) նշանակվում է բոլոր հնարավոր (n, k)-տեղափոխությունների քանակը: P(n, k) = = :

## ԿՐԿՆՈՒԹՅՈՒՆՆԵՐՈՎ ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ

Կրկնություններով տեղափոխությունները տարրերի սովորական վերադասավորումներ չեն: Այստեղ տարրի կրկնությունը կարելի է մեկնաբանել որպես ընտրված տարրի վերադարձ նախնական բազմություն, որից հետո այն ստանում է հաջորդ քայլում նորից ընտրվելու հնարավորություն: Մյուս մեկնաբանությունն այն է, որ յուրաքանչյուր տարրից ունենք անսահմանափակ քանակությամբ կրկնօրինակներ, որոնք կարող են ընտրվել հաջորդ տեղափոխություններում:

## (n, k)-ՏԵՂԱՓՈԽՈՒԹՅՈՒՆՆԵՐ ԿՐԿՆՈՒԹՅՈՒՆՆԵՐՈՎ

Կրկնություններով (n, k)-տեղափոխությունները այն (n, k)-տեղափոխություններն են, որոնց տարրերը կարող են կրկնվել:

U(n, k)-ով նշանակվում է կրկնություններով բոլոր հնարավոր (n, k)-տեղափոխությունների քանակը: U(n, k)=nk:

## ԶՈՒԳՈՐԴՈՒԹՅՈՒՆՆԵՐ

(n, k)-զուգորդություններ են կոչվում այն (n, k)-ընտրությունները, որոնց տարրերը չեն կրկնվում, և տարրերի դասավորման կարգը նշանակություն չունի, այսինքն՝ նրա դասավորությունը փոխելով, նոր զուգորդություն չենք ստանա: Բոլոր հնարավոր (n, k)-զուգորդությունների քանակը նշանակվում է C(n, k)-ով, իսկ երբեմն -ով կամ -ով: C(n, k) = = = :

## ԶՈՒԳՈՐԴՈՒԹՅՈՒՆ ԿՐԿՆՈՒԹՅՈՒՆՆԵՐՈՎ

(n, k)-զուգորդություն կրկնություններով կոչվում է այն (n, k)-ընտրությունը, որի տարրերը կարող են կրկնվել, իսկ տարրերի դասավորման կարգը նշանակություն չունի: Կրկնություններով (n, k)-զուգորդությունների քանակը նշանակվում է V(n, k)-ով:

V(n, k) = :

## ՏԱՐՐԱԿԱՆ ՆՈՒՅՆՈՒԹՅՈՒՆՆԵՐ

1. =
2. = +
3. + + … + = 2n

## ՆՅՈՒՏՈՆԻ ԵՐԿԱՆԴԱՄ

=:

## ՊԱՍԿԱԼԻ ԵՌԱՆԿՅՈՒՆ

Հաշվելով Նյուտոնի երկանդամային գործակիցները n = 0, 1, … արժեքների համար և տեղադրելով իրար տակ, կստանանք հետևյալ պատկերը, որը կոչվում է Պասկալի եռանկյուն.



# ԲԱԺԻՆ 3

## ԳՐԱՖՆԵՐ

Մաթեմատիկայում գրաֆը մի շարք օբյեկտներեի վերացական ներկայացումն է, որտեղ մի քանի զույգ օբյեկտներ կապված են հղումներով: Փոխկապակցված օբյեկտները ներկայացվում են մաթեմատիկական աբստրակցիաների միջոցով, որոնք կոչվում են գագաթներ և կողեր: Սխեմատիկ տեսքով գրաֆը կարելի է պատկերել որպես մի շարք կետերի և դրանք միացնող գծերի կամ կորերի միջոցով: Գրաֆները մեկն են այն օբյեկտներից, որոնք ուսումնասիրվում են դիսկրետ մաթեմատիկայում:

## ԳՐԱՖ

Դիցուք V-ն ոչ դատարկ բազմություն է, P2(V)-ն V-ի բոլոր հնարավոր զույգերի բազմությունն է E⊆P2(V): Այդ դեպքում G = (V, E) զույգը կոչվում է գրաֆ:

## ԳՐԱՖՆԵՐԻ ԳԱԳԱԹՆԵՐ ԵՎ ԿՈՂԵՐ

Եթե G = (V, E)-ն գրաֆ է, ապա V-ի տարրերը կոչվում են G գրաֆի գագաթներ, իսկ E-ի տարրերը՝ կողեր: Գրաֆի {x, y}∈E կողի համար սովորաբար օգտագործվում է xy կրճատ նշանակումը:

## ԳՐԱՖԻ ԵՐԿՐԱՉԱՓԱԿԱՆ ՄԵԿՆԱԲԱՆՈՒԹՅՈՒՆԸ

G = (V, E) գրաֆը հարմար է պատկերել նկարի ձևով, որը բաղկացած է կետերից (որոնք համապատասխանում են գրաֆի գագաթներին) և այդ կետերի որոշ զույգերը միացնող գծերից (որոնք համապատասխանում են գրաֆի կողերին):

## ԿՈՂԻ ԾԱՅՐԱԿԵՏԵՐ, ԿՑՈՒԹՅՈՒՆ, ՀԱՐԵՎԱՆ ԳԱԳԹՆԵՐ ԵՎ ՀԱՐԵՎԱՆ ԿՈՂԵՐ

Եթե xy-ը կող է (այսինքն xy∈E), ապա x և y գագաթները կոչվում են xy կողի ծայրակետեր և ասում են, որ xy կողը կից է x և y գագաթներին, կամ x, y գագաթները կից են xy կողին: Միևնույն ժամանակ, xy∈E կողի առկայության դեպքում ասում են, որ x և y գագաթները հարևան են: Երկու կողեր կոչվում են հարևան, եթե ունեն ընդհանուր ծայրակետ:

## ԳԱԳԱԹԻ ՀԱՐԵՎԱՆՈՒԹՅՈՒՆ

G = (V, E) գրաֆի v գագաթի հարևանությունը v գագաթին հարևան բոլոր գագաթների բազմությունն է և նշանակվում է N(v):

## ՀԱՆԳՈՒՅՑՆԵՐ ԵՎ ԶՈՒԳԱՀԵՌ ԿՈՂԵՐ

Հանգույց կոչվում է aa տեսքի կողը, այսինքն՝ այնպիսի կողը, որը միացնում է նույն a գագաթն ինքն իր հետ: Զուգահեռ կոչվում են մեկից ավելի այնպիսի կողերը, որոնք միացնում են միևնույն եկու գագաթները:

## ՊԱՐԶ ԿԱՄ ՍՈՎՈՐԱԿԱՆ ԳՐԱՖՆԵՐ

Գրաֆը, որը չունի հանգույցներ և զուգահեռ կողեր, կոչվում է պարզ գրաֆ կամ սովորական գրաֆ:

## ՄՈՒԼՏԻԳՐԱՖ

Զուգահեռ կողեր ունեցող գրաֆը կոչվում է մուլտիգրաֆ:

## ՊՍԵՎԴՈԳՐԱՖ

Հանգույցներ թույլատրող գրաֆը անվանում են պսեվդոգրաֆ:

## ԿՈՂՄՆՈՐՈՇՎԱԾ ԳՐԱՖ

Եթե գրաֆի կողերը կարգավորված զույգեր են, ապա գրաֆը կոչվում է կողմնորոշված:

## ՀԻՊԵՐԳՐԱՖ

Դիցուք V-ն ոչ դատարկ բազմություն է, P(V)-ն V-ի բոլոր հնարավոր ենթաբազմությունների բազմությունն է և E⊆P(V): Այդ դեպքում G=(V, E) զույգը կոչվում է հիպերգրաֆ: Հիպերգրաֆի կողերը երկրաչափորեն նույնքան հեշտ չէ պատկերել, որքան գրաֆի կողերը:

## 

## ՆՇԱԳՐՎԱԾ ԿԱՄ ԿՇՌՎԱԾ ԳՐԱՖ

Եթե գրաֆի կողերին վերագրված են նիշեր կամ թվեր, ապա այդպիսի գրաֆը կոչվում է նշագրված կամ կշռված գրաֆ:

## ՀԱՐԵՎԱՆՈՒԹՅԱՆ ՄԱՏՐԻՑ

Գագաթների V={v1, v2, …, vn} բազմություն և կողերի E={e1, e2, …, eq} բազմություն G = (V, E) գրաֆի հարևանության մատրիցն n×n չափսի քառակուսային բուլյան մատրից է՝

## ԿՑՈՒԹՅԱՆ ՄԱՏՐԻՑ

Գագաթների V={v1, v2, …, vn} բազմություն և կողերի E={e1, e2, …, eq} բազմություն ունեցող G = (V, E) գրաֆի կցության մատրիցը n×q չասի բուլյան մատրից է՝

## ՀԱՐԵՎԱՆՈՒԹՅԱՆ ՑՈՒՑԱԿՆԵՐ

Եթե գրաֆի գագաթները համարակալված են 1, 2, …, n թվերով, ապա գրաֆի հարևանության ցուցակներն իրենցից ներկայացնում են N[1], N[2], …, N[n] տեսքի մասսիվներ, որտեղ N[i]-ն i-րդ գագաթի հարևան գագաթների համարների ցուցակն է: Հարևանության ցուցակների գրաված հիշողության ծավալը n+2q է:

## ԳՐԱՖԻ ՆԵՐԿԱՅԱՑՄԱՆ ԵՂԱՆԱԿՆԵՐԸ

Հայտնի են գրաֆի ներկայացման հետևյալ հիմնական եղանակները՝

1. գրաֆի գագաթների և կողերի բազմություններ,
2. երկրաչափական պատկեր, որը բաղկացած է կետերից (գագաթներից) և այդ կետերի որոշ զույգերը միացնող գծերից (կողերից),
3. հարևանության մատրից,
4. կցության մատրից,
5. գագաթների հարևանության ցուցակներ:

## ՀԱՎԱՍԱՐ ԳՐԱՖՆԵՐ

G և H գրաֆները կոչվում են հավասար, եթե V(G)=V(H) և E(G)=E(H):

## ԻԶՈՄՈՐՖ ԳՐԱՖՆԵՐ

G և H գրաֆները կոչվում են իզոմորֆ (նշանակվում է G≅H), եթե V(G) և V(H) բազմությունների միջև գոյություն ունի այնպիսի f փոխմիարժեք համապատասխանություն, որ xy∈E(G) այն և միայն այն ժամանակ, երբ f(x)f(y)∈E(H):

## ԵՆԹԱԳՐԱՖ, ԿՄԱԽՔԱՅԻՆ ԵՆԹԱԳՐԱՖ ԵՎ ԾՆՎԱԾ ԵՆԹԱԳՐԱՖ

H գրաֆը կոչվում է G գրաֆի ենթագրաֆ, եթե V(H)⊆V(G) և E(H)⊆E(G): G գրաֆի H ենթագրաֆը կոչվում է կմախքային, եթե V(H)=V(G): G գրաֆի H ենթագրաֆը կոչվում է S գագաթների բազմությամբ ծնված ենթագրաֆ, կամ ուղղակի ծնված ենթագրաֆ, եթե V(H)=S⊆V(G) և E(H)={vu | v,u∈S և vu∈E(G)}:

## ԼՐԻՎ ԳՐԱՖ

G գրաֆը կոչվում է լրիվ, եթե նրա ցանկացած երկու գագաթներ հարևան են: n գագաթների լրիվ գրաֆը նշանակվում է Kn: Հաճախ K3 գրաֆը անվանում են եռանկյուն:

## ԳԱԳԱԹԻ ԱՍՏԻՃԱՆ, ՄԵԿՈՒՍԱՑՎԱԾ ԵՎ ԿԱԽՎԱԾ ԳԱԳԱԹՆԵՐ

G գրաֆի v գագաթի աստիճանը դա v գագաթի բոլոր հարևան գագաթների քանակն է և նշանակվում է d(v): Եթե d(v)=0, ապա v-ն կոչվում է մեկուսացված գագաթ, իսկ եթե d(v)=1, ապա կոչվում է կախված գագաթ:

## ՆՎԱԶԱԳՈՒՅՆ ԵՎ ԱՌԱՎԵԼԱԳՈՒՅՆ ԱՍՏԻՃԱՆՆԵՐ

δ(G)-ով և Δ(G)-ով նշանակվում են գրաֆի գագաթների համապատասխանաբար նվազագույն և առավելագույն աստիճանները:

## ՀԱՄԱՍԵՌ ԳՐԱՖ

Համասեռ (r-համասեռ) է կոչվում այն գրաֆը, որի բոլոր գագաթների աստիճանները իրար հավասար են (հավասար են r-ի): 3-համասեռ գրաֆը հաճախ անվանում են նաև խորանարդ գրաֆ:

## ԵՐԿԿՈՂՄԱՆԻ ԳՐԱՖ

G գրաֆը կոչվում է երկկողմանի կամ երկմասնյա, եթե նրա գագաթների բազմությունը կարելի է տրոհել այնպիսի երկու V1 և V2 չհատվող ենթաբազմությունների, որ միևնույն ենթաբազմության գագաթները զույգ առ զույգ հարևան չեն: Եթե երկկողմանի գրաֆի V1 բազմության կամայական գագաթ հարևան է V2 բազմության կամայական գագաթին, ապա գրաֆը կոչվում է լրիվ երկկոմանի և նշանակվում է Km,n, որտեղ m=, n=:

## ԳՐԱՖԱՅԻՆ ՀԱՋՈՐԴԱԿԱՆՈՒԹՅՈՒՆ

d1, d2, …, dn թվերի հավաքածուն կոչվում է գրաֆային, եթե գոյություն ունի այնպիսի գրաֆ, որի համար d1, d2, …, dn հաջորդականությունը նրա գագաթների աստիճանների հաջորդականությունն է:

## ԱՆԿԱԽ ԳԱԳԱԹՆԵՐ, ԱՆԱԽՈՒԹՅԱՆ ԹԻՎ, ԱՆԿԱԽ ԿՈՂԵՐ

G գրաֆի զույգ առ զույգ ոչ հարևան գագաթները կոչվում են անկախ գագաթներ: Գրաֆում անկախ գագաթների առավելագույն քանակը կոչվում է գրաֆի անկախության թիվ և նշանակվում է α(G): Գրաֆի զույգ առ զույգ ոչ հարևան կողերը կոչվում են անկախ կողեր:

## ԶՈՒԳԱԿՑՈՒՄ, ԿԱՏԱՐՅԱԼ ԶՈՒԳԱԿՑՈՒՄ, 1-ՖԱԿՏՈՐ

Գրաֆում անկախ կողերի բազմությունը անվանում են նաև զուգակցում: Զուգակցումը կոչվում է մաքսիմալ, եթե նրա կողերի քանակը հնարավոր ամենամեծն է: Կատարյալ զուգակցումը դա այն զուգակցումն է, որն ընդգրկում է գրաֆի բոլոր գագաթները: Կատարյալ զուգակցումը երբեմն անվանում են նաև 1-ֆակտոր: Այլ կերպ ասած, կատարյալ զուգակցումը (կամ 1-ֆակտոր) դա գրաֆի 1-համասեռ կմախքային ենթագրաֆն է:

ԳՐԱՖԻ ՏՐԱՄԱԳԻԾ, ԿԵՆՏՐՈՆ ԵՎ ՇԱՌԱՎԻՂ

Գրաֆի d(G) տրամագիծը և r(G) շառավիղը համապատասխանաբար սահմանվում են հետևյալ կերպ.

d(G) = maxmaxd(u, v), r(G) = minmaxd(u, v)

v

u

v

u

Այլ կերպ ասած, գրաֆի տրամագիծը դա գրաֆի իրարից ամենահեռու երկու գագաթների հեռավորությունն է: Դիցուք r(v)-ն ցույց է տալիս տված v գագաթից ամենահեռու գագաթի հեռավորությունը: v գագաթը կոչվում է գագաթի կենտրոն, եթե v գագաթի համար r(v) թիվը հնարավոր ամենափոքԿրն է: Այդ դեպքում, գրաֆի շառավիղը կլինի կենտրոնից ամենահեռու գագաթի հեռավորությունը:

ՑԻԿԼՈՄԱՏԻԿ ԹԻՎ

Կամայական G = (V, E) գրաֆ բնութագրվում է c(G) = - + 1 բնութագրիչով, որը անվանում են G գրաֆի ցիկլոմատիկ թիվ:

# ԳԼՈՒԽ 2

## ԳՈՐԾՆԱԿԱՆ ՄԱՍ: ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ, ԿՈՄԲԻՆԱՏՈՐԻԿԱ ԵՎ ԳՐԱՖՆԵՐ

# ԲԱԺԻՆ 1

ԲՈՒԼՅԱՆ ՖՈՒՆԿՑԻԱՆԵՐ

y= | x → y ⊕ z ↓ (x | y)

Տրված բուլյան ֆունկցիայի համար ստանալ.

1. Ֆունկցիայի իսկության աղյուսակը և իսկության վեկտորը
2. Հեռացնել ֆունկցիայի ֆիկտիվ փոփոխականները
3. Ստանալ բուլյան ֆունկցիայի ԿԴՆՁ և ԿԿՆՁ
4. Ստանալ բուլյան ֆունկցիայի Ժեգալկինի բազմանդամը
5. Կառուցել բուլյան ֆունկցիայի ֆտս, որի բարդությունը չի գերազանցում 6-ը
6. Գրել ծրագիր, որի արդյունքում կստանանք ֆունկցիայի ԿԴՆՁ և ԿԿՆՁ-ն

## ԻՍԿՈՒԹՅԱՆ ԱՂՅՈՒՍԱԿ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | a |  | b | e | c | d | f |
|  | x y z |  | x ∨ |  | x → y | x | y | | b | c ⊕ z | d ↓ e |
| 0 | 0 0 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 0 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 2 | 0 1 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | 0 1 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 4 | 1 0 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 5 | 1 0 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 6 | 1 1 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 7 | 1 1 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Իսկության աղյուսակը ունի հետևյալ տեսքը.

Առաջին քայլով կատարվել է (y-ի ժխտում) գործողությունը, որից հետո կատարվել է x ∨ (x կամ ) գործողությունը և ստացվածը նշանակվել է a-ով, որից հետո կատարվել է (a-ի ժխտում գործողությունը): Այնուհետև կատարվել է x → y (x-ից հետևում է y-ը) գործողությունը և ստացվածը նշանակվել է b-ով: Դրանից հետո կատարվել է x | y (ոչ x կամ ոչ y) գործողությունը և արդյունքը նշանակվել է e-ով: Այնուհետև կատարվել է | b (ոչ a-ի ժխտում կամ ոչ b) գործողությունը և նշանակվել է c-ով: Դրանից հետո կատարվել է (c ⊕ z) (c-ի և z-ի մոդուլ 2-ով գումարում) գործողությունը և արդյունքը նշանակվել է d-ով: Վերջին քայլում կատարվել է d ↓ e (ոչ d, ոչ e) գործողությունը, որի արդյունքը իսկության վեկտորն է:

## ԻՍԿՈՒԹՅԱՆ ՎԵԿՏՈՐ

Իսկության վեկտորը կլինի՝ f(x, y, z) = (0000 0001)

## ԷԱԿԱՆ ԵՎ ԿԵՂԾ ՓՈՓՈԽԱԿԱՆՆԵՐ

X փոփոխականի էական կամ կեղծ լինելը ստուգվել է հետևյալ կերպ.

Վերցվել է այն տողերը, որտեղ x փոփոխականն իր արժեքը փոխում է 0-ից 1 իսկ y և z փոփոխականները իրենց արժեքները այդ տողերում չեն փոխում՝ 0-ն մնում է 0, 1-ը` 1: Այնուհետև այդ տողերում ստուգվում են իսկության վեկտորի ընդունած արժեքները: Եթե իսկության վեկտորը գոնե մի զույգի համար փոխում է իր արժեքը, ապա x-ը էական փոփոխական է, հակառակ դեպքում՝ կեղծ: 3 փոփոխականի ֆունկցիայի համար այդ ստուգվող զույգերը 4-ն են (0-4; 1-5; 2-6; 3-7): Պարզվել է, որ x-ը էական փոփոխական է, քանի որ 3-7 զույգի համար իսկության վեկտորը փոխել է իր արժեքը, 0-ից դարձել է 1:

Y փոփոխականի համար վերցվել են այն տողերը, որտեղ y փոփոխականը փոխել է իր արժեքը իսկ x և z փոփոխականները իրենց արժեքներն այդ տողերում չեն փոխել: Այդ զույգերը ⋅կլինեն 4-ը (0-2; 1-3; 4-6; 5-7): Ստուգումից պարզվել է, որ y-ը էական փոփոխական է, քանի որ 5-7 զույգում իսկության վեկտորն իր արժեքը փոխել է՝ 0-ից դարձել է 1:

Z փոփոխականի համար զույգեր են կազմվել այն տողերից, որտեղ z-ն իր արժեքը փոխել է իսկ x և y փոփոխականները մնացել են նույնը: Այդ զույգերը՝ ինչպես նախորդ դեպքերում եղել են 4-ը (0-1; 2-3; 4-5; 6-7): Պարզվել է, որ z փոփոխականը նույնպես էական է, քանի որ վերջին զույգում իսկության վեկտորն իր արժեքը փոխել է:

Այսքանից պարզ է դառնում, որ կեղծ (ֆիկտիվ) փոփոխականներ չկան, և բնականաբար ոչ մի փոփոխական չի հեռացվել:

## ԿԱՏԱՐՅԱԼ ԴԻԶՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎ (ԿԴՆՁ)

ԿԴՆՁ-ն գտնելու համար վերցվել են այն տողերը, որտեղ իսկության վեկտորն ընդունել է 1 արժեք, այնուհետև այդ տողում փոփոխականների ընդունած արժեքները վերցվել են կամ ուղիղ ձևով կամ ժխտած (եթե այդ տողում փոփոխականը 1 արժեք է ընդունել վերցվել է ուղիղ ձևով, հակառակ դեպքում ժխտված) և հետո կատարվել է դրանց կոնյունկցիա: Եթե իսկության վեկտորը մեկից ավելի տողերում է ընդունում 1 արժեք, ապա այդ տողի համար նույնպես կատարվում է վերը նշված գործողությունը, և մյուսների հետ միացվում է դիզյունկցիայով: Այս օրինակում իսկության մենակ 7-րդ տողում է ընդունել 1 արժեք: Վերցվել է այդ տողի x փոփոխականը, քանի որ այն ընդունել է 1 արժեք, վերցվել է ուղիղ ձևով: y և z փոփոխականները այդ տողում նույնպես ընդունել են 1 արժեք, հետևաբար ԿԴՆՁ-ն եղել է x⋅y⋅z:

ԿԴՆՁ՝ x⋅y⋅z:

## ԿԱՏԱՐՅԱԼ ԿՈՆՅՈՒՆԿՏԻՎ ՆՈՐՄԱԼ ՁԵՎ (ԿԿՆՁ)

ԿԿՆՁ-ն գտնելու համար վերցվել են այն տողերը, որտեղ իսկության վեկտորն ընդունել է 0 արժեք, այնուհետև այդ տողում փոփոխականների ընդունած արժեքները վերցվել են կամ ուղիղ ձևով կամ ժխտած (եթե այդ տողոում փոփոխականը 0 արժեք է ընդունել վերցվել է ուղիղ ձևով, հակառակ դեպքում ժխտված) և հետո կատարվել է դրանց դիզյունկցիա: Եթե իսկության վեկտորը մեկից ավելի տողերում է ընդունում 0 արժեք, ապա այդ տողի համար նույնպես կատարվում է վերը նշված գործողությունը, և մյուսների հետ միացվում է կոնյունկցիայով: Այս օրինակում իսկության վեկտորը 0 արժեք ընդունել է 0, 1, 2, 3, 4, 5, 6 տողերում:

ԿԿՆՁ-ն ընդունել է հետևյալ տեսքը՝ (x ∨ y ∨z) ⋅ (x ∨ y ∨ ) ⋅ (x ∨ ∨ z) ⋅ (x ∨ ∨ ) ⋅ ( ∨ y ∨ z) ⋅ ( ∨ y ∨ ) ⋅ ( ∨ ∨ z):

Ստացված ԿԿՆՁ-ն կարելի է կրճատել հետևյալ կերպ.

1. Առաջին և երկրորդ փակագծերը բացել միմյանց հետ, կստացվի՝ xx ∨ xy ∨ x ∨ yx ∨ yy ∨ y ∨ zx ∨ zy ∨ z=x ∨ xy ∨ x ∨ yx ∨ y ∨ y ∨ zx ∨ zy = x ∨ y ∨ y ∨ zx ∨ zy = x ∨ y ∨ zy = x ∨ y:
2. Երրորդ և չորրորդ փակագծերը բացել միմյանց հետ, կստացվի՝ xx ∨ x ∨ x ∨ x ∨ ∨ ∨ zx ∨ z ∨ z = x ∨ x ∨ x ∨ z ∨ ∨ ∨ zx ∨ z = x ∨ ∨ ∨ z = x ∨ :
3. Հինգերորդ և վեցերորդ փակագծերը բացել միմյանց հետ, կստացվի՝ ∨ y ∨ ∨ y ∨ yy ∨ y ∨ z ∨ zy ∨ z = ∨ y ∨ ∨ y ∨ y ∨ y ∨ z ∨ zy = ∨ y ∨ y ∨ zy = ∨ y:
4. Դրանից հետո 1-ին և 2-րդ կետերում ստացված արդյունքները բացել միմյանց հետ, կստացվի՝ xx ∨ x ∨ yx ∨ y = x ∨ x ∨ yx = x:
5. 3-րդ և 4-րդ կետերում ստացված արդյունքը բացել միմյանց հետ, կստացվի ՝ x ∨ xy = xy:
6. Վերջին քայլով 5-րդ կետում ստացվածը պետք է բացել սկզբնական ԿԿՆՁ-ի յոթերորդ փակագծի հետ, կստացվի՝ xy ∨ xy ∨ xyz = xyz:

Վերջնական ԿԿՆՁ-ն ստացվեց xyz:

## ԺԵԳԱԼԿԻՆԻ ԲԱԶՄԱՆԴԱՄ

Այս օրինակում Ժեգալկինի բազմանդամը ստացվել է երկու ճանապարհով՝ առաջինը ԿԿՆՁ-ի միջոցով, երկրորդը՝ անորոշ գործակիցների մեթոդով:

1. Այս դեպքում ԿԿՆՁ-ն ձևափոխվում է այնպես, որպիսի պարունակի միայն կոնյունկցիա և մոդուլ 2-ով գումարում: Քանի որ ԿԿՆՁ-ն կրճատելուց հետո ստացվել էր xyz, ապա Ժեգալկինի բազմանդամը նույնպես լինում է xyz:
2. Անորոշ գործակիցների մեթոդով Ժեգալկինի բազմանդամը ստացվել է հետևյալ կերպ.

β0K0 ⊕ β1K1 ⊕ β2K2 ⊕ β3K3 ⊕ β4K4 ⊕ β5K5 ⊕ β6K6 ⊕ β7K7, որտեղ

K0 = 1 K2 = y K4 = x K6 = xy

K1 = z K3 = xy K5 = xz K7=xyz

|  |  |
| --- | --- |
| x y z | F(x,y,z) |
| 0 0 0 | 0 |
| 0 0 1 | 0 |
| 0 1 0 | 0 |
| 0 1 1 | 0 |
| 1 0 0 | 0 |
| 1 0 1 | 0 |
| 1 1 0 | 0 |
| 1 1 1 | 1 |



Ժեգալկինի բազմանդամը կլինի β7⋅xyz = 1⋅xyz = xyz:

## ՖՈՒՆԿՑԻՈՆԱԼ ՏԱՐՐԵՐԻ ՍԽԵՄԱ (ՖՏՍ)

Տվյալ ֆունկցիայի ֆունկցիոնալ տարրերի սխեման կառուցվել է երկու ձևով.

1. Ըստ ԿԿՆՁ-ի, ԿԿՆՁ-ն ստացվել էր xyz, հետևաբար ֆտս-ն կընդունի հետևյալ տեսքը.



1. ՖՏՍ-ն կառուցվել է նաև ձևափոխելով տրված ֆունկցիան: Ձևափոխումը կատարվել է հետևյալ կերպ.

| x → y ⊕ z ↓ (x | y) = x ∨ ∨ → y ⊕ z ↓ (x | y) = 1 ∨ → y ⊕ z ↓ (x | y) = 1 ∨ y ∨ y ⊕ z ↓ (x | y) = 1 ⊕ z ↓ (x | y) = ↓ (x | y):

Այսպիսով ֆտս-ն կընդունի հետևյալ տեսքը․



Ինչպես երևում է, երկու դեպքում էլ ֆտս-ի բարդությունը չի գերազանցում 6-ը:

# ԲԱԺԻՆ 2

ԿՈՄԲԻՆԱՏՈՐԻԿԱ

## ԽՆԴԻՐ 1

2.1.6) Դիցուք՝ S-ը բոլոր քառանիշ թվերի բազմությունն է, որոնց մեջ մասնակցում են 0, 1, 2, 3, և 6 թվանշանները:

ա) Ի՞նչ հզորություն ունի S-ը:

բ) S-ի մեջ քանի՞ թվեր կրկնվող թվանշաններ չեն պարունակում:

գ) Նախորդ խնդրում քանի՞ զույգ թվեր կան:

դ) Երկրորդ խնդրում 4000-ից մեծ քանի՞ թվեր կան:

ա) Ունենք 5 թվանշան, որոնցից մեկը 0-ն է: Քանի որ քառանիշ թիվը չի կարող սկսել 0-ով, ուրեմն թվի առաջին թվանշանի համար ունենք 4 տարբերակ՝ 1, 2, 3, 6: Թվի մյուս թվանշանների համար սահմանափակում չունենք, ուրեմն տրված թվանշաններով կարելի է կազմել 4\*5\*5\*5 = 500 հատ քառանիշ թիվ:

P = 4\*5\*5\*5 = 500

բ) Ինչպես նշվեց նախորդ կետում, քառանիշ թիվը չի կարող սկսել 0-ով, ուրեմն առաջին դիրքի համար ունենք 4 տարբերակ՝ 1, 2, 3, 6: Քանի որ սրանցից մեկն արդեն օգտագրծվել է, իսկ 0-ն դեռ ոչ, ուրեմն երկրորդ դիրքում կարելի է օգտագործել 0-ն և մնացած 3-ից որևէ մեկը, այսինքն ունենք 4 տարբերակ: Երկրորդ դիրքում կունենանք ընտրելու 3 հնարավորություն, իսկ առաջինում՝ 2: Կունենանք 4\*4\*3\*2 = 96 հատ չկրկնվող թվանշաններով քառանիշ թիվ:

N = 4\*4\*3\*2 = 96:

գ) Թիվը զույգ է, եթե նրա վերջին թվանշանը զույգ է: Այս դեպքում զույգ թվանշաններն են՝ 0, 2, 6: Այսինքն կունենանք չկրկնվող թվանշաններով կազմված 3 զույգ թիվ:

դ) Որպիսի քառանիշ թիվը լինի 4000-ից մեծ, պետք է առաջին թվանշանը լինի 6: Եթե 6 թիվն արդեն օգտագործել ենք, երկրորդ դիրքի համար կմնա 4 տարբերակ՝ 0, 1, 2, 3: Երկրորդ դիրքում կունենանք ընտրելու 3 հնարավորություն, իսկ առաջինում՝ 2: Այսինքն կունենանք 4\*3\*2 = 24 հատ թիվ, որոնք կրկնվող թվանշաններ չունեն և մեծ են 4000-ից:

N1 = 4\*3\*2 = 24:

## ԽՆԴԻՐ 2

2.5.8) Քանի՞ ձևերով կարելի է հանել 5 խաղաթուղթ, որոնցից յուրաքանչյուրում ուղիղ 2-ը մի կարգի են, իսկ մյուս 2-ը այլ կարգի (օրինակ, 2 տուզ և 2 թագավոր):

Ունենք 36 խաղաթուղթ, որոնց մեջ կան 4 հատ ամեն կարգից: Կարգերն են՝ 6-նոցներ, 7-նոցներ, 8-նոցներ, 9-նոցներ, 10- նոցներ,վալետներ, թագուհիներ, թագավորներ, տուզեր: Ամեն մի կարգից ունենք 4 ռանգ՝ սիրտ, քափ, խաչ, ղառ: Սկզբում 2 նույն կարգի խաղաթուղթ հանելու տարբերակները 9\*: Արդեն ընտրելուց հետո կմնա 8 կարգ, կլինի 8\*: Հինգերորդ խաղաթուղը ընտրելու համար ունենք 38-8 = 28 տարբերակ:

Ընդհանուր կլինի 9\* \* 8\* \* 28 = 72576 տարբերակ:

# ԲԱԺԻՆ 3

ԳՐԱՖՆԵՐ

## 

## 

1. Գտնել տրված գրաֆի համար կցության և հարևանության մատրիցները և գրել ծրագիր մատրիցների հաշվման և արտածման համար:
2. Գտնել տրված երկու գագաթների համար նվազագույն ճանապարհը գտնելու ալգորիթմը և ծրագիրը:
3. Գրել ծրագիր, որը կարտածի գրաֆի max-իմում գագաթի աստիճան ունեցող գագաթը և նրա աստիճանի չափը:
4. Տրված գրաֆի համար ստանալ բոլոր գագաթների աստիճանները համապատասխան գագաթներով, և արտածել գագաթի աստիճաններն աճման կարգով: Գրել ծրագիր:
5. Գրել ծրագիր, որը կպարզի գրաֆը ունի կախված կամ մեկուսացված գագաթ, եթե այո տպի հաղորդագրություն:
6. Գտնել տրված գրաֆի տրամագիծը և շառավիղը:
7. Գտնել տրված գրաֆի ցիկլոմատիկ թիվը:
8. Գտնել տրված գրաֆում անկախության թիվը:
9. Տրված գրաֆի կցության և հարևանության մատրիցների ստացումը.

* Գրաֆի հարևանության մատրիցը կլինի.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | V0 | V1 | V2 | V3 | V4 | V5 | V6 |
| V0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| V1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| V2 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| V3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| V4 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| V5 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| V6 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |

Ստացվել է 7 տող և 7 սյուն, քանի որ գրաֆն ունի 7 գագաթ.

V0 գագաթը կապված է V1, V2, V3, V4 և V6 գագաթների հետ, այդ իսկ պատճառով մատրիցի V0 տողում և սյունում այդ գագաթների համար ստացվել է 1 արժեք:

V1 գագաթը կապված է V0, V2, և V5 գագաթների հետ, դրա համար մատրիցի V1 տողում և սյունում այդ գագաթների համար ստացվել է 1 արժեք, իսկ մնացած գագաթների համար 0 արժեք:

V2 գագաթը կապված է V0, V1, V3, V4 և V5 գագաթների հետ, դրա համար մատրիցի V2 տողում և սյունում այդ գագաթների համար ստացվել է 1 արժեք, իսկ մնացած գագաթների համար՝ 0:

V3 գագաթը կապված է V0 և V2 գագաթների հետ, դրա համար մատրիցի V3 տողում և սյունում այդ 2 գագաթների համար ստացվել է 1 արժեք, իսկ մնացած 5 գագաթների համար՝ 0:

V4 գագաթը կապված է V0, V2 և V5 գագաթների հետ, այդ պատճառով մատրիցի V4 տողում և սյունում այդ գագաթների համար ստացվել է 1 արժեք, իսկ մնացած գագաթների համար 0 արժեք:

V5 գագաթը կապված է V1, V2, V4 և V6 գագաթների հետ, այդ պատճառով մատրիցի V5 տողում և սյունում այդ գագաթների համար ստացվել է 1 արժեք, իսկ մնացած գագաթների համար՝ 0:

V6 գագաթը կապված է V0 և V5 գագաթների հետ, այդ պատճառով մատրիցի V6 տողում և սյունում այդ 2 գագաթների համար ստացվել է 1 արժեք, իսկ մնացած 5 գագաթների համար 0 արժեք:

* Գրաֆի կցության մատրիցը կլինի.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | V0 | V1 | V2 | V3 | V4 | V5 | V6 |
| Е0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Е1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Е2 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Е3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Е4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Е5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Е6 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| Е7 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Е8 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Е9 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| Е10 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Е11 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Ստացվել է 7 սյուն և 12 տող, քանի որ գրաֆն ունի 7 գագաթ և 12 կողմ:

Ըստ նշանակման ՝

1. V0 և V1 գագաթները միանում են Е0 կողմով, դրա համար մատրիցի Е0 տողում V0 և V1 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
2. V0 և V2 գագաթները միանում են Е1 կողմով, դրա համար մատրիցի Е1 տողում V0 և V2 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
3. V0 և V3 գագաթները միանում են Е2 կողմով, դրա համար մատրիցի Е2 տողում V0 և V3 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
4. V0 և V2 գագաթները միանում են Е1 կողմով, դրա համար մատրիցի Е1 տողում V0 և V2 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
5. V0 և V4 գագաթները միանում են Е3 կողմով, դրա համար մատրիցի Е3 տողում V0 և V4 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
6. V0 և V6 գագաթները միանում են Е4 կողմով, դրա համար մատրիցի Е4 տողում V0 և V6 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
7. V1 և V2 գագաթները միանում են Е5 կողմով, դրա համար մատրիցի Е5 տողում V1 և V2 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
8. V1 և V5 գագաթները միանում են Е6 կողմով, դրա համար մատրիցի Е6 տողում V1 և V5 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
9. V2 և V3 գագաթները միանում են Е7 կողմով, դրա համար մատրիցի Е7 տողում V2 և V3 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
10. V2 և V4 գագաթները միանում են Е8 կողմով, դրա համար մատրիցի Е8 տողում V2 և V4 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
11. V2 և V5 գագաթները միանում են Е9 կողմով, դրա համար մատրիցի Е9 տողում V2 և V5 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
12. V4 և V5 գագաթները միանում են Е10 կողմով, դրա համար մատրիցի Е10 տողում V4 և V5 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
13. V5 և V6 գագաթները միանում են Е11 կողմով, դրա համար մատրիցի Е11 տողում V5 և V6 սյուների համար ստացվել է 1 արժեք, իսկ մնացած սյուների համար 0 արժեք:
14. V3 գագաթից V5 գագաթ կարելի է անցնել հետևյալ ճանապարհներով՝

* V3 → V2 → V5
* V3 → V2 → V4 → V5
* V3 → V2 → V1 → V5
* V3 → V2 → V4 → V5
* V3 → V2 → V0 → V4 → V5
* V3 → V2 → V4 → V0 → V1 → V5
* V3 → V0 → V1 → V5
* V3 → V0 → V1 → V2 → V5
* V3 → V0 → V1 → V2 → V4 → V5
* V3 → V0 → V6 → V5
* V3 → V0 → V4 → V5
* V3 → V0 → V4 → V2 → V5

Քանի որ գրաֆի յուրաքանչյուր 2 գագաթներ միացնող ճանապարհի երկարությունը հավասար է 1-ի, ապա ակնհայտ երևում է, որ V3 գագաթից V5 գագաթ տանող ամենակարճ ճանապարհը V3 → V2 → V5-ն, իսկ եթե յուրաքանչյուր կողմի համար տրված լիներ կոնկրետ երկարություն, ապա պետք է այդ գագաթները միացնող բոլոր հնարավոր տարբերակները նկատի ունենայինք, և դրանից հետո նոր գտնվեր ամենակարճ ճանապարհը:

1. Գագաթի աստիճանները գտնվում են ըստ հարևանության մատրիցի, յուրքանչյուր գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Հարևանության մատրիցից երևում է, որ V0 գագաթն ունի 5 հարևան, V1 գագաթը՝ 3 հարևան, V2 գագաթը՝ 5 հարևան, V3 գագաթը՝ 2 հարևան, V4 գագաթը՝ 3 հարևան, V5 գագաթը՝ 4 հարևան և V6 գագաթը՝ 2 հարևան:

Ստացվեց, որ մաքսիմում գագաթի աստիճան ունեն V0 և V2 գագաթները, դրանց արժեքը հավասար է 5-ի:

1. Գագաթի աստիճաններն աճման կարգով դասավորելու համար նախ անհրաժեշտ է գտնել բոլոր գագաթների գագաթի աստիճանը:

V0 գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Ուրեմն d(V0)=5:

V1 գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Ուրեմն d(V1)=3:

V2 գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Ուրեմն d(V2)=5:

V3 գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Ուրեմն d(V3)=2:

V4 գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Ուրեմն d(V4)=3:

V5 գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Ուրեմն d(V5)=4:

V6 գագաթի գագաթի աստիճանը հավասար է նրան հարևան գագաթների թվին: Ուրեմն d(V6)=2:

Ունենք 5, 3, 5, 2, 3, 4, 2 թվերը: Այս թվերն աճման կարգով կդասավորվի հետևյալ կերպ.

2, 2, 3, 3, 4, 5, 5:

1. Քանի որ գագաթի աստիճաններում չունենք այնպիսի գագաթ, որի արժեքը հավասար է 1-ի, ուրեմն գրաֆը չունի կախված գագաթ: Չունենք նաև մեկուսացած գագաթ, քանի որ գագաթի աստիճաններում չկա այնպիսին, որի արժեքը հավասար է 0-ի:
2. Ըստ սահմանման տրամագիծը հավասար է երկու գագաթները միացնող ճանապարհներից ամենաերկարին: Այս գրաֆում V3 և V5 գագաթները միացնող ամենաերկար ճանապարհը հետևյալն է՝

* V3 🡪 V0 🡪 V1 🡪 V2 🡪 V4 🡪 V5
* V3 🡪 V0 🡪 V4 🡪 V2 🡪 V1 🡪 V5
* V3 🡪 V2 🡪 V1 🡪 V0 🡪 V4 🡪 V5
* V3 🡪 V2 🡪 V1 🡪 V0 🡪 V6 🡪 V5
* V3 🡪 V2 🡪 V4 🡪 V0 🡪 V1 🡪 V5
* V3 🡪 V2 🡪 V4 🡪 V0 🡪 V6 🡪 V5

Ինչպես տեսնում ենք տրված երկու գագաթները միանում են հինգ քայլով, ուրեմն գրաֆի տրամագիծը հավասար է հինգի:

Ըստ սահմանման գրաֆի շառավիղը հավասար է տրված երկու գագաթները միացնող ամենակարճ ճանապարհի երկարությանը: Տրված V3 և V5 գագաթները միացնող ամենակարճ ճանապարհը V3 🡪 V2 🡪 V5 է, որի երկարությունը հավասար է երկուսի: Ուրեմն շառավիղը հավասար է երկուսի:

1. Ցիկլոմատիկ թիիվը գտնելու համար անհրաժեշտ է գրաֆի կողմերի քանակից հանել գրաֆի գագթների քանակը և ստացվածին գումարել 1` E - V + 1: Տրված գրաֆում ունենք 7 գագաթ 12 կողմ, ուրեմն ցիկլոմատիկ թիվը հավասար կլինի 6-ի:
2. Գրաֆի անկախության թիվը հավասար է զույգ առ զույգ ոչ հարևան գագաթների քանակին: Սակայն հաշվի չենք առնում գագաթն ինքն իր հետ հարևան լինելը: Այսինքն անկախության թիվը գտնելու համար կարող ենք հարևանության մատրիցի 0-ների քանակից հանենք գագաթների քանակը և ստացվածը բաժանենք 2-ի (քանի որ զույգ առ զույգ հարևան չլինելն է մեկով ավելացնում անկախության թիվը): Տրված գրաֆի համար անկախության թիվը հավասար է 9-ի:

# ԾՐԱԳԻՐ

#include <iostream>

**using** **namespace** std;

**void** **kdndz**()

{

{

**int** hamar;

**do** {

cout << endl;

cout << "ԸՆՏՐԵԼ ՀՐԱՄԱՆԸ" << endl;

cout << "2.2 ՓՈՓՈԽԱԿԱՆԻ ՖՈՒՆԿՑԻԱ " << endl;

cout << "3.3 ՓՈՓՈԽԱԿԱՆԻ ՖՈՒՆԿՑԻԱ" << endl;

cout << "4.4 ՓՈՓՈԽԱԿԱՆԻ ՖՈՒՆԿՑԻԱ" << endl;

cout << "5.ԵԼՔ" << endl;

cin >> hamar;

{

**if** (hamar == **3**)

{

cout << "x | " << "y|" << " z" << endl;

cout << "0 | " << "0|" << " 0" << endl;

cout << "0 | " << "0|" << " 1" << endl;

cout << "0 | " << "1|" << " 0" << endl;

cout << "0 | " << "1|" << " 1" << endl;

cout << "1 | " << "0|" << " 0" << endl;

cout << "1 | " << "0|" << " 1" << endl;

cout << "1 | " << "1|" << " 0" << endl;

cout << "1 | " << "1|" << " 1" << endl;

**int** x[] = { **0**,**0**,**0**,**0**,**1**,**1**,**1**,**1** };

**int** y[] = { **0**,**0**,**1**,**1**,**0**,**0**,**1**,**1** };

**int** z[] = { **0**,**1**,**0**,**1**,**0**,**1**,**0**,**1** };

**int** l = **0**, k = **0**, b, q = **0**, u = **0**, t = **0**;

**int** f[**8**], zang[**8**], hat = **0**;

cout << "f(x,y,z)= ";

**for** (**int** i = **0**; i < **8**; i++)

**do**

{

cin >> f[i];

**if** (f[i] == **1**) hat++;

zang[i] = f[i];

} **while** (f[i] < **0** || f[i] > **1**);

**int** krj = **0**;

**if** (f[**0**] == **0** && f[**1**] == **0** && f[**2**] == **0** && f[**3**] == **0** && f[**4**] == **0** && f[**5**] == **0** && f[**6**] == **0** && f[**7**] == **1**) krj = **1**;

**for** (**int** i = **0**; i < **4**; i++)

**if** (f[i] == f[i + **4**]) l++;

**if** (l == **4**)

{

q = **1**;

cout << "x-ը ֆիկտիվ է" << endl;

**for** (**int** i = **0**; i < **4**; i++)

{

f[i] = f[i + **4**];

cout << f[i];

}

}

**else** k++;

**int** m = **0**;

**for** (**int** i = **0**; i < **2**; i++)

**if** (f[i] == f[i + **2**]) m++;

**for** (**int** i = **4**; i < **6**; i++)

**if** (f[i] == f[i + **2**]) m++;

**if** (m == **4**)

{

cout << "y-ը ֆիկտիվ է" << endl;

**if** (q == **0**) **for** (**int** i = **0**; i < **4**; i++)

**if** (i < **2**)

{

f[i] = f[i + **2**];

cout << f[i];

}

**else**

{

f[i] = f[i + **2**];

cout << f[i];

}

**else**

**for** (**int** i = **0**; i < **3**; i++)

{

**if** (i < **2**)

{

f[i] = f[i + **2**];

cout << f[i];

}

}

u = **1**;

}

**else** k++;

**int** n = **0**;

**for** (**int** i = **0**; i < **1**; i++) **if** (f[i] == f[i + **1**]) n++;

**for** (**int** i = **2**; i < **3**; i++) **if** (f[i] == f[i + **1**]) n++;

**for** (**int** i = **4**; i < **5**; i++) **if** (f[i] == f[i + **1**]) n++;

**for** (**int** i = **6**; i < **7**; i++) **if** (f[i] == f[i + **1**]) n++;

**if** (n == **4**)

{

cout << "z-ը ֆիկտիվ է" << endl;

**if** (q == **0** && u == **0**)

**for** (**int** i = **0**; i < **4**; i++)

{

**if** (i == **0**)

{

f[i] = f[i];

cout << f[i] << " ";

}

**if** (i == **1**)

{

f[i] = f[i + **1**];

cout << f[i] << " ";

}

**if** (i == **2**)

{

f[i] = f[i + **2**];

cout << f[i] << " ";

}

**if** (i == **3**)

{

f[i] = f[i + **3**];

cout << f[i] << " ";

}

}

**if** (q == **1** && u == **0**)

**for** (**int** i = **0**; i < **2**; i++)

{

**if** (i == **0**)

{

f[i] = f[i];

cout << f[i] << " ";

}

**if** (i == **1**)

{

f[i] = f[i + **1**];

cout << f[i] << " ";

}

}

**if** (q == **0** && u == **1**)

**for** (**int** i = **0**; i < **2**; i++)

{

**if** (i == **0**)

{

f[i] = f[i];

cout << f[i] << " ";

}

**if** (i == **1**)

{

f[i] = f[i + **4**];

cout << f[i] << " ";

}

}

t = **1**;

}

**else** k++;

**if** (k == **3**) cout << "ֆիկտիվ փոփոխական չկա" << endl;

**if** (k == **0**) cout << "բոլորը ֆիկտիվ են" << endl;

cout << endl;

cout << "ԿԴՆՁ` ";

**int** a = **0**, hat1 = **0**;

**for** (**int** i = **0**; i < **8**; i++)

**if** (f[i] == **0**) a++;

**if** (a == **8**) cout << "ԿԴՆՁ չկա";

**for** (**int** i = **0**; i < **8**; i++)

{

**if** (zang[i] == **1**)

{

hat1++;

**if** (x[i] == **1**) cout << "x"; **else** cout << "x'";

**if** (y[i] == **1**) cout << "y"; **else** cout << "y'";

**if** (z[i] == **1**) cout << "z"; **else** cout << "z'";

**if** (hat > hat1) cout << " V ";

}

}

cout << endl;

cout << "ԿԿՆՁ` ";

**int** h = **0**;

**for** (**int** i = **0**; i < **8**; i++)

{

**if** (zang[i] == **0**)

{

cout << "(";

**if** (x[i] == **0**) cout << "x V "; **else** cout << "x' V ";

**if** (y[i] == **0**) cout << "y V "; **else** cout << "y' V ";

**if** (z[i] == **0**) cout << "z )"; **else** cout << "z' )";

}

**else**

**if** (f[i] == **1**) h++;

**if** (h == **8**) cout << "ԿԿՆՁ չկա";

}

**if** (krj == **1**)

cout << "=(XYX' V XYY' V XYZ)=XYZ" << endl;

}

}

**if** (hamar == **2**)

{

cout << "x | " << "y" << endl;

cout << "0 | " << "0" << endl;

cout << "0 | " << "1" << endl;

cout << "1 | " << "0" << endl;

cout << "1 | " << "1" << endl;

**int** en = **0**, n = **0**, qu = **0**, el = **0**, ka = **0**;

**int** c[] = { **0**,**0**,**1**,**1** };

**int** v[] = { **0**,**1**,**0**,**1** };

**int** funk[**4**];

**int** zangvac[**4**];

cout << "f(x,y)= ";

**for** (**int** i = **0**; i < **4**; i++)

**do**

{

cin >> funk[i];

**if** (funk[i] == **1**) n++;

zangvac[i] = funk[i];

} **while** (funk[i] < **0** || funk[i] > **1**);

**for** (**int** i = **0**; i < **2**; i++)

**if** (funk[i] == funk[i + **2**]) el++;

**if** (el == **2**)

{

qu = **1**;

cout << "x-ը ֆիկտիվ է" << endl;

**for** (**int** i = **0**; i < **2**; i++)

{

funk[i] = funk[i + **2**];

cout << funk[i] << " ";

}

cout << endl;

}

**else** ka++;

**int** ele = **0**, qyu = **0**;

**if** (funk[**0**] == funk[**1**]) ele++;

**if** (funk[**2**] == funk[**3**]) ele++;

**if** (ele == **2**)

{

qyu = **1**;

cout << "y-ը ֆիկտիվ է" << endl;

**if** (qu == **1**)

**for** (**int** i = **0**; i < **1**; i++) funk[i] = funk[i];

**else**

{

**for** (**int** i = **0**; i < **2**; i++)

{

**if** (i == **0**)

funk[i] = funk[i]; **else** funk[i] = funk[i + **1**];

cout << funk[i] << " ";

}

cout << endl;

}

}

**else**

ka++;

**int** qan = **0**;

**if** (qyu == **1** && qu == **1**) cout << "բոլորը ֆիկտիվ են" << endl;

**if** (qyu == **0** && qu == **0**) cout << "ֆիկտիվ փոփոխական չկա" << endl;

cout << "ԿԴՆՁ` ";

**for** (**int** i = **0**; i < **4**; i++) **if** (zangvac[i] == **1**)

{

en++;

**if** (c[i] == **1**) cout << "x"; **else** cout << "x'";

**if** (v[i] == **1**) cout << "y"; **else** cout << "y'";

**if** (n > en) cout << " V ";

}

**else**

**if** (zangvac[i] == **0**) qan++;

**if** (qan == **4**) cout << "ԿԴՆՁ չկա";

**int** qanak1 = **0**;

cout << endl;

cout << "ԿԿՆՁ` ";

**for** (**int** i = **0**; i < **4**; i++) **if** (zangvac[i] == **0**)

{

cout << "(";

**if** (c[i] == **0**) cout << "x V"; **else** cout << "x' V";

**if** (v[i] == **0**) cout << " y)"; **else** cout << " y')";

}

**else**

**if** (zangvac[i] == **1**) qanak1++;

**if** (qanak1 == **4**) cout << "ԿԿՆՁ չկա";

}

**if** (hamar == **4**)

{

cout << "x | " << "y|" << " z|" << " p" << endl;

cout << "0 | " << "0|" << " 0|" << " 0" << endl;

cout << "0 | " << "0|" << " 0|" << " 1" << endl;

cout << "0 | " << "0|" << " 1|" << " 0" << endl;

cout << "0 | " << "0|" << " 1|" << " 1" << endl;

cout << "0 | " << "1|" << " 0|" << " 0" << endl;

cout << "0 | " << "1|" << " 0|" << " 1" << endl;

cout << "0 | " << "1|" << " 1|" << " 0" << endl;

cout << "0 | " << "1|" << " 1|" << " 1" << endl;

cout << "1 | " << "0|" << " 0|" << " 0" << endl;

cout << "1 | " << "0|" << " 0|" << " 1" << endl;

cout << "1 | " << "0|" << " 1|" << " 0" << endl;

cout << "1 | " << "0|" << " 1|" << " 1" << endl;

cout << "1 | " << "1|" << " 0|" << " 0" << endl;

cout << "1 | " << "1|" << " 0|" << " 1" << endl;

cout << "1 | " << "1|" << " 1|" << " 0" << endl;

cout << "1 | " << "1|" << " 1|" << " 1" << endl;

cout << "f(x,y,z,p)=";

**int** syun1[] = { **0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**,**1**,**1**,**1**,**1**,**1**,**1**,**1**,**1** };

**int** syun2[] = { **0**,**0**,**0**,**0**,**1**,**1**,**1**,**1**,**0**,**0**,**0**,**0**,**1**,**1**,**1**,**1** };

**int** syun3[] = { **0**,**0**,**1**,**1**,**0**,**0**,**1**,**1**,**0**,**0**,**1**,**1**,**0**,**0**,**1**,**1** };

**int** syun4[] = { **0**,**1**,**0**,**1**,**0**,**1**,**0**,**1**,**0**,**1**,**0**,**1**,**0**,**1**,**0**,**1** };

**int** funk\_4[**16**];

**int** zang\_4[**16**];

**int** em = **0**, b;

**int** tp\_x = **0**;

**for** (**int** i = **0**; i < **16**; i++)

**do**

{

cin >> funk\_4[i];

**if** (funk\_4[i] == **1**) em++;

zang\_4[i] = funk\_4[i];

} **while** (funk\_4[i] < **0** || funk\_4[i] > **1**);

**int** x\_fik = **0**;

**for** (**int** i = **0**; i < **8**; i++)

**if** (funk\_4[i] == funk\_4[i + **8**]) x\_fik++;

**if** (x\_fik == **8**)

{

tp\_x = **1**;

cout << "x-ը ֆիկտիվ է" << endl;

**for** (**int** i = **0**; i < **8**; i++)

{

funk\_4[i] = funk\_4[i + **8**];

cout << funk\_4[i] << " ";

}

cout << endl;

}

**int** y\_fik = **0**;

**int** tp\_y = **0**;

**for** (**int** i = **0**; i < **4**; i++)

**if** (funk\_4[i] == funk\_4[i + **4**]) y\_fik++;

**for** (**int** i = **8**; i < **12**; i++)

**if** (funk\_4[i] == funk\_4[i + **4**]) y\_fik++;

**if** (y\_fik == **8**)

{

cout << "y-ը ֆիկտիվ է" << endl;

tp\_y = **1**;

**if** (tp\_x == **0** && tp\_y == **1**)

{

**for** (**int** i = **0**; i < **4**; i++)

{

cout << funk\_4[i] << " ";

}

**for** (**int** i = **8**; i < **12**; i++)

{

cout << funk\_4[i] << " ";

}

}

}

**if** (tp\_x == **1** && tp\_y == **1**)

**for** (**int** i = **0**; i < **4**; i++)

{

cout << funk\_4[i] << " ";

}

cout << endl;

**int** z\_fik = **0**;

**int** tp\_z = **0**;

**for** (**int** i = **0**; i < **2**; i++)

**if** (funk\_4[i] == funk\_4[i + **2**]) z\_fik++;

**for** (**int** i = **4**; i < **6**; i++)

**if** (funk\_4[i] == funk\_4[i + **2**]) z\_fik++;

**for** (**int** i = **8**; i < **10**; i++)

**if** (funk\_4[i] == funk\_4[i + **2**]) z\_fik++;

**for** (**int** i = **12**; i < **14**; i++)

**if** (funk\_4[i] == funk\_4[i + **2**]) z\_fik++;

**if** (z\_fik == **8**)

{

cout << "z-ը ֆիկտիվ է" << endl;

tp\_z = **1**;

**if** (tp\_x == **0** && tp\_y == **0** && tp\_z == **1**)

{

**for** (**int** i = **0**; i < **2**; i++) cout << funk\_4[i] << " ";

**for** (**int** i = **4**; i < **6**; i++) cout << funk\_4[i] << " ";

**for** (**int** i = **8**; i < **10**; i++) cout << funk\_4[i] << " ";

**for** (**int** i = **12**; i < **14**; i++) cout << funk\_4[i] << " ";

}

**if** (tp\_x == **1** && tp\_y == **0** && tp\_z == **1**)

{

**for** (**int** i = **0**; i < **2**; i++) cout << funk\_4[i] << " ";

**for** (**int** i = **4**; i < **6**; i++) cout << funk\_4[i] << " ";

}

**if** (tp\_x == **0** && tp\_y == **1** && tp\_z == **1**)

{

**for** (**int** i = **0**; i < **2**; i++) cout << funk\_4[i] << " ";

**for** (**int** i = **8**; i < **10**; i++) cout << funk\_4[i] << " ";

}

**if** (tp\_x == **1** && tp\_y == **1** && tp\_z == **1**)

{

**for** (**int** i = **0**; i < **2**; i++) cout << funk\_4[i] << " ";

}

}

cout << endl;

**int** p\_fik = **0**;

**int** tp\_p = **0**;

**for** (**int** i = **0**; i < **1**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;

**for** (**int** i = **2**; i < **3**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;

**for** (**int** i = **4**; i < **5**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;;

**for** (**int** i = **6**; i < **7**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;

**for** (**int** i = **8**; i < **9**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;

**for** (**int** i = **10**; i < **11**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;

**for** (**int** i = **12**; i < **13**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;

**for** (**int** i = **14**; i < **15**; i++)

**if** (funk\_4[i] == funk\_4[i + **1**]) p\_fik++;

**if** (p\_fik == **8**)

{

cout << "p-ն ֆիկտիվ է" << endl;

tp\_p = **1**;

**if** (tp\_x == **0** && tp\_y == **0** && tp\_z == **0**)

**for** (**int** i = **0**; i < **16**; i++)

**if** (i % **2** == **0**) cout << funk\_4[i] << " ";

**if** (tp\_x == **0** && tp\_y == **0** && tp\_z == **1**)

{

cout << funk\_4[**0**] << " ";

cout << funk\_4[**4**] << " ";

cout << funk\_4[**8**] << " ";

cout << funk\_4[**12**] << " ";

}

**if** (tp\_x == **0** && tp\_y == **1** && tp\_z == **0**)

{

cout << funk\_4[**0**] << " ";

cout << funk\_4[**2**] << " ";

cout << funk\_4[**8**] << " ";

cout << funk\_4[**10**] << " ";

}

**if** (tp\_x == **0** && tp\_y == **1** && tp\_z == **1**)

{

cout << funk\_4[**0**] << " ";

cout << funk\_4[**8**] << " ";

}

**if** (tp\_x == **1** && tp\_y == **0** && tp\_z == **0**)

{

**for** (**int** i = **0**; i < **8**; i++)

**if** (i % **2** == **0**) cout << funk\_4[i];

}

**if** (tp\_x == **1** && tp\_y == **0** && tp\_z == **1**)

{

cout << funk\_4[**0**] << " ";

cout << funk\_4[**4**] << " ";

}

**if** (tp\_x == **1** && tp\_y == **1** && tp\_z == **0**)

{

cout << funk\_4[**0**] << " ";

cout << funk\_4[**2**] << " ";

}

**if** (tp\_x == **1** && tp\_y == **1** && tp\_z == **1**)

cout << "բոլորը ֆիկտիվ են" << endl;

}

**if** (tp\_x == **0** && tp\_y == **0** && tp\_z == **0** && tp\_p == **0**)

cout << "ֆիկտիվ փոփոխական չկա" << endl;

cout << endl;

cout << "ԿԴՆՁ` ";

**int** diz\_chka = **0**;

**for** (**int** i = **0**; i < **16**; i++)

**if** (funk\_4[i] == **0**) diz\_chka++;

**if** (diz\_chka == **16**) cout << "ԿԴՆՁ չկա" << endl;

**int** a = **0**, m = **0**;

**for** (**int** i = **0**; i < **16**; i++)

{

**if** (zang\_4[i] == **1**)

{

m++;

**if** (syun1[i] == **1**) cout << "x"; **else** cout << "x'";

**if** (syun2[i] == **1**) cout << "y"; **else** cout << "y'";

**if** (syun3[i] == **1**) cout << "z"; **else** cout << "z'";

**if** (syun4[i] == **1**) cout << "p"; **else** cout << "p'";

**if** (em > m) cout << " V ";

}

**else**

**for** (**int** i = **0**; i < **16**; i++)

**if** (funk\_4[i] == **0**) a++;

}

cout << endl;

cout << "ԿԿՆՁ` ";

**int** aa = **0**, eme = **0**;

**for** (**int** i = **0**; i < **16**; i++)

{

**if** (zang\_4[i] == **0**)

{

eme++;

cout << "(";

**if** (syun1[i] == **0**) cout << "x V"; **else** cout << " x' V";

**if** (syun2[i] == **0**) cout << " y V"; **else** cout << " y' V";

**if** (syun3[i] == **0**) cout << " z V"; **else** cout << " z' V";

**if** (syun4[i] == **0**) cout << " p)"; **else** cout << " p')";

}

**else**

**for** (**int** i = **0**; i < **16**; i++)

**if** (funk\_4[i] == **1**) aa++;

**if** (aa == **16**) cout << "ԿԿՆՁ չկա";

}

}

} **while** (hamar != **5**);

}

}

**void** **graph**()

{

{

**int** gagatner, koxmer, a;

**do**

{

cout << "ԳԱԳԱԹՆԵՐ=";

cin >> gagatner;

} **while** (gagatner < **0**);

**do**

{

cout << "ԿՈՂՄԵՐ=";

cin >> koxmer;

} **while** (koxmer < **0**);

**int** matrix[**30**][**30**];

**int** matrix1[**30**][**30**];

**for** (**int** i = **0**; i < gagatner; i++)

**for** (**int** j = **0**; j < gagatner; j++)

matrix[i][j] = **0**;

**for** (**int** i = **0**; i < koxmer; i++)

**for** (**int** j = **0**; j < gagatner; j++)

matrix1[i][j] = **0**;

**int** x, y;

**for** (**int** i = **0**; i < koxmer; i++)

{

cin >> x >> y;

matrix[x][y] = **1**;

matrix[y][x] = **1**;

matrix1[i][x] = **1**;

matrix1[i][y] = **1**;

}

**int** zroner = **0**;

**for** (**int** i = **0**; i < gagatner; i++)

**for** (**int** j = **0**; j < gagatner; j++)

**if** (matrix[i][j] == **0**) zroner++;

cout << endl;

cout << "ՀԱՐԵՎԱՆՈՒԹՅԱՆ ՄԱՏՐԻՑ";

cout << endl;

**for** (**int** i = **0**; i < gagatner; i++)

{

**for** (**int** j = **0**; j < gagatner; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << endl;

cout << "ԿՑՈՒԹՅԱՆ ՄԱՏՐԻՑ";

cout << endl;

**for** (**int** i = **0**; i < koxmer; i++)

{

**for** (**int** j = **0**; j < gagatner; j++)

{

cout << matrix1[i][j] << " ";

}

cout << endl;

}

cout << "ԱՆԿԱԽՈՒԹՅԱՆ ԹԻՎ = " << (zroner - gagatner) / **2** << endl;

cout << endl;

**int** qan; **int** d[**100**];

**for** (**int** i = **0**; i < gagatner; i++)

{

qan = **0**;

**for** (**int** j = **0**; j < gagatner; j++)

**if** (matrix[i][j] == **1**)

qan++;

d[i] = qan;

cout << "V" << i << "-ն ունի " << qan << " հարևան" << endl;

**if** (qan == **0**)

cout << "V" << i << "-ն մեկուսացված" << endl;

**if** (qan == **1**)

cout << "V" << i << "-ն կախված է" << endl;

}

**for** (**int** i = **0**; i < gagatner; i++)

cout << d[i] << " ";

cout << endl;

**int** max = d[**0**];

**for** (**int** i = **1**; i < gagatner; i++)

**if** (d[i] >= max)

max = d[i];

**for** (**int** i = **0**; i < gagatner; i++)

**if** (d[i] == max)

cout << "V" << i << "-ն ունի մեծագույն գագաթի աստիճան` " << max << endl;

cout << endl;

cout << "գագաթի աստիճաններն աճման կարգով՝ ";

**int** index[**50**] = { **0** }, ii = **0**;

**for** (**int** i = **0**; i < gagatner; i++)

**for** (**int** j = i + **1**; j < gagatner; j++)

**if** (d[j] < d[i])

{

a = d[i];

d[i] = d[j];

d[j] = a;

}

**for** (**int** i = **0**; i < gagatner; i++)

cout << d[i] << " ";

cout << endl;

cout << endl;

**int** cikl\_tiv;

cikl\_tiv = abs(koxmer) - abs(gagatner) + **1**;

cout << "ցիկլոմատիկ թիվ = " << cikl\_tiv << endl;

**int** w;

cout << "կարճագույն ճանապարհը երկու գագաթների միջև" << endl;

**int** gag1, gag2, h, k;

**int** v, u, b;

**do**

{

cout << "գագաթ 1=";

cin >> gag1;

} **while** (gag1<**0** || gag1>gagatner);

**do**

{

cout << "գագաթ 2=";

cin >> gag2;

} **while** (gag2<**0** || gag2>gagatner);

**int** qan1 = **0**;

**if** (matrix[gag1][gag2] == **1**)

{

cout << "1 քայլով՝ V" << gag1 << " --> V" << gag2 << endl;

qan1++;

w = **1**;

}

**else**

**for** (**int** i = **0**; i < gagatner; i++)

**if** (matrix[gag1][i] == **1**)

{

h = i;

**if** (matrix[h][gag2] == **1**)

{

cout << "2 քայլով` V" << gag1 << " --> V" << h << " --> V" << gag2 << endl;

qan1++;

w = **2**;

}

}

**if** (qan1 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (j == h) **continue**;

**if** (matrix[h][j] == **1**)

{

k = j;

**if** (matrix[k][gag2] == **1**)

{

cout << "3 քայլով` V" << gag1 << " --> V" << h

<< " --> V" << k << " --> V" << gag2 << endl;

qan1++;

w = **3**;

}

}

}

}

}

**if** (qan1 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1**)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1**)

{

v = l;

**if** (matrix[v][gag2] == **1**)

{

cout << "4 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k

<< " --> V" << l << " --> V" << gag2 << endl;

qan1++;

w = **4**;

}

}

}

}

}

}

}

**if** (qan1 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1**)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1**)

{

v = l;

**for** (**int** q = **0**; q < gagatner; q++)

**if** (matrix[v][q] == **1**)

{

u = q;

**if** (matrix[u][gag2] == **1**)

{

cout << "5 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k << " --> V"

<< v << " --> V" << u << " --> V" << gag2 << endl;

qan1++;

w = **5**;

}

}

}

}

}

}

}

}

**if** (qan1 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1**)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1**)

{

v = l;

**for** (**int** q = **0**; q < gagatner; q++)

**if** (matrix[v][q] == **1**)

{

u = q;

**for** (**int** g = **0**; g < gagatner; g++)

{

**if** (matrix[u][g] == **1**)

{

b = g;

**if** (matrix[b][gag2] == **1**)

{

cout << "6 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k

<< " --> V" << v << " --> V" << u << " --> V" << b << " --> V" << gag2 << endl;

qan1++;

w = **6**;

}

}

}

}

}

}

}

}

}

}

**int** p;

**if** (qan1 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1**)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1**)

{

v = l;

**for** (**int** q = **0**; q < gagatner; q++)

**if** (matrix[v][q] == **1**)

{

u = q;

**for** (**int** g = **0**; g < gagatner; g++)

{

**if** (matrix[u][g] == **1**)

{

b = g;

{

**for** (**int** t = **0**; t < gagatner; t++)

{

**if** (matrix[b][t] == **1**)

{

p = t;

**if** (matrix[p][gag2] == **1**)

{

cout << "7 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k

<< " --> V" << v << " --> V" << u

<< " --> V" << b << " --> V" << p << " --> V" << gag2 << endl;

qan1++;

w = **7**;

}

}

}

}

}

}

}

}

}

}

}

}

}

cout << "շառավիղ = " << w << endl;

cout << endl;

cout << endl;

cout << "տրամագիծ" << endl;

**int** qan2 = **0**; **int** ww;

**if** (qan2 == **0**)

{

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1** && j != h && j != gag1)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1** && l != h && l != k && l != gag1)

{

v = l;

**for** (**int** q = **0**; q < gagatner; q++)

**if** (matrix[v][q] == **1** && q != h && q != k && q != v && q != gag1)

{

u = q;

**for** (**int** g = **0**; g < gagatner; g++)

{

**if** (matrix[u][g] == **1** && g != h && g != k && g != v && g != u && g != gag1)

{

b = g;

{

**for** (**int** t = **0**; t < gagatner; t++)

{

**if** (matrix[b][t] == **1** && t != h && t != k && t != v && t != u && t != b && t != gag1)

{

p = t;

**if** (matrix[p][gag2] == **1** && gag2 != h && gag2 != k && gag2 != v

&& gag2 != u && gag2 != b && gag2 != p)

{

cout << "7 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k

<< " --> V" << v << " --> V" << u

<< " --> V" << b << " --> V" << p << " --> V" << gag2 << endl;

qan2++;

ww = **7**;

}

}

}

}

}

}

}

}

}

}

}

}

}

}

**if** (qan2 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1** && j != h && j != gag1)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1** && l != h && l != k && l != gag1)

{

v = l;

**for** (**int** q = **0**; q < gagatner; q++)

**if** (matrix[v][q] == **1** && q != h && q != k && q != v && q != gag1)

{

u = q;

**for** (**int** g = **0**; g < gagatner; g++)

{

**if** (matrix[u][g] == **1** && g != h && g != k && g != v && g != u && g != gag1)

{

b = g;

**if** (matrix[b][gag2] == **1** && gag2 != h && gag2 != k && gag2 != v && gag2 != u)

{

cout << "6 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k

<< " --> V" << v << " --> V" << u << " --> V" << b << " --> V" << gag2 << endl;

qan2++;

ww = **6**;

}

}

}

}

}

}

}

}

}

}

**if** (qan2 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1** && j != h && j != gag1)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1** && l != h && l != k && l != gag1)

{

v = l;

**for** (**int** q = **0**; q < gagatner; q++)

**if** (matrix[v][q] == **1** && q != h && q != k && q != v && q != gag1)

{

u = q;

**if** (matrix[u][gag2] == **1** && gag2 != h && gag2 != k && gag2 != v)

{

cout << "5 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k << " --> V"

<< v << " --> V" << u << " --> V" << gag2 << endl;

qan2++;

ww = **5**;

}

}

}

}

}

}

}

}

**if** (qan2 == **0**)

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1** && j != h && j != gag1)

{

k = j;

**for** (**int** l = **0**; l < gagatner; l++)

{

**if** (matrix[k][l] == **1** && l != h && l != k && l != gag1)

{

v = l;

**if** (matrix[v][gag2] == **1** && gag2 != h && gag2 != k)

{

cout << "4 քայլով՝ V" << gag1 << " --> V" << h << " --> V" << k << " --> V"

<< l << " --> V" << gag2 << endl;

qan2++;

ww = **4**;

}

}

}

}

}

}

}

**if** (qan2 == **0**)

{

**for** (**int** i = **0**; i < gagatner; i++)

{

**if** (matrix[gag1][i] == **1**)

{

h = i;

**for** (**int** j = **0**; j < gagatner; j++)

{

**if** (matrix[h][j] == **1** && j != h && j != gag1)

{

k = j;

**if** (matrix[k][gag2] == **1** && gag2 != h)

{

cout << "3 քայլով՝ V" << gag1 << " --> V" << h

<< " --> V" << k << " --> V" << gag2 << endl;

qan2++;

ww = **3**;

}

}

}

}

}

}

**if** (qan2 == **0**)

{

**for** (**int** i = **0**; i < gagatner; i++)

**if** (matrix[gag1][i] == **1**)

{

h = i;

**if** (matrix[h][gag2] == **1**)

{

cout << "2 քայլով` V" << gag1 << " --> V" << h << " --> V" << gag2 << endl;

qan2++;

ww = **2**;

}

}

}

cout << "տրամագիծ = " << ww << endl;

}

}

**int** **main**()

{

**int** yntrel;

**do**

{

cout << " ՄԵՆՅՈՒ " << endl;

cout << "1.ԿԴՆՁ ԵՎ ԿԿՆՁ" << endl;

cout << "2.ԳՐԱՖ" << endl;

out << "3.ԵԼՔ" << endl;

cout << "ՀՐԱՄԱՆ = ";

cin >> yntrel;

**switch** (yntrel)

{

**case** **1**:

{

kdndz();

**break**;

}

**case** **2**:

{

graph();

**break**;

}

**case** **3**:

{

**break**;

}

**default:**

{

cout << "ՍԽԱԼ ՀՐԱՄԱՆ" << endl;

**break**;

}

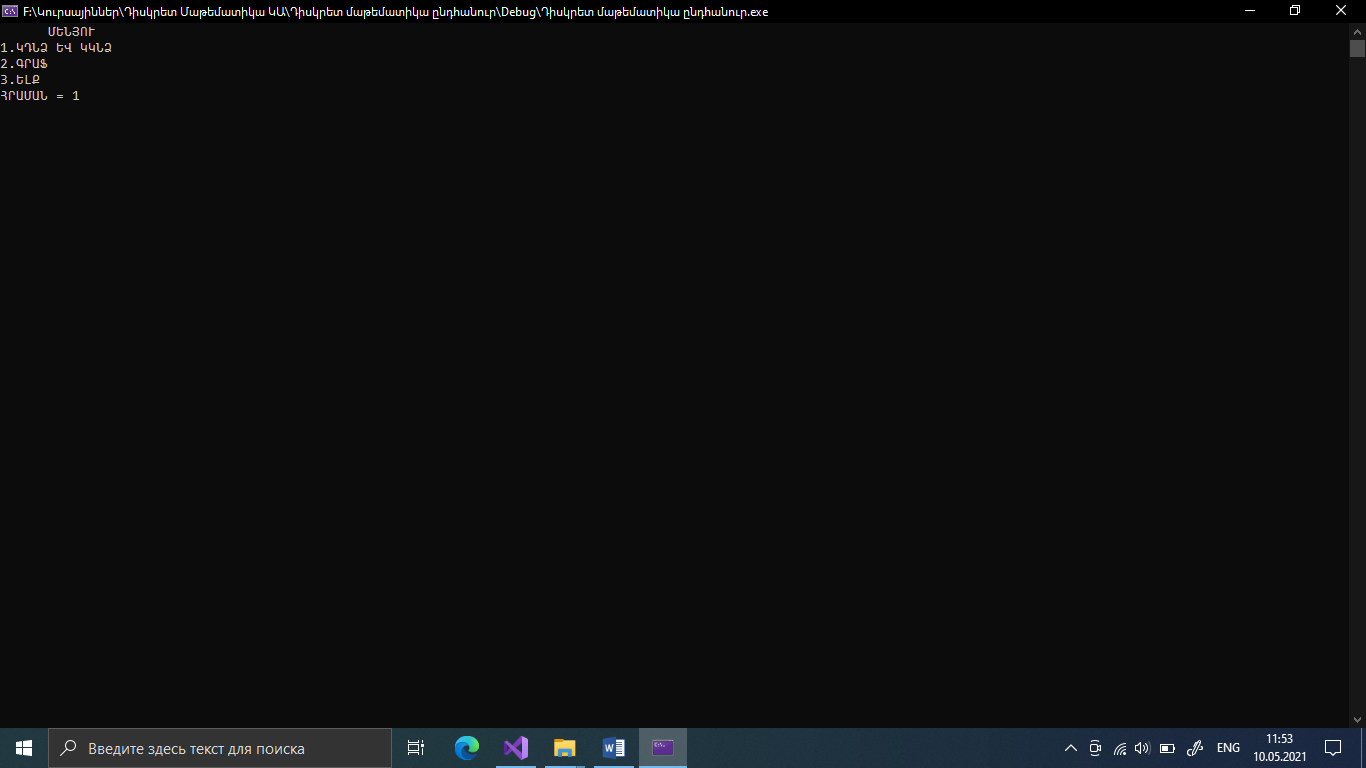
}

} **while** (yntrel != **3**);

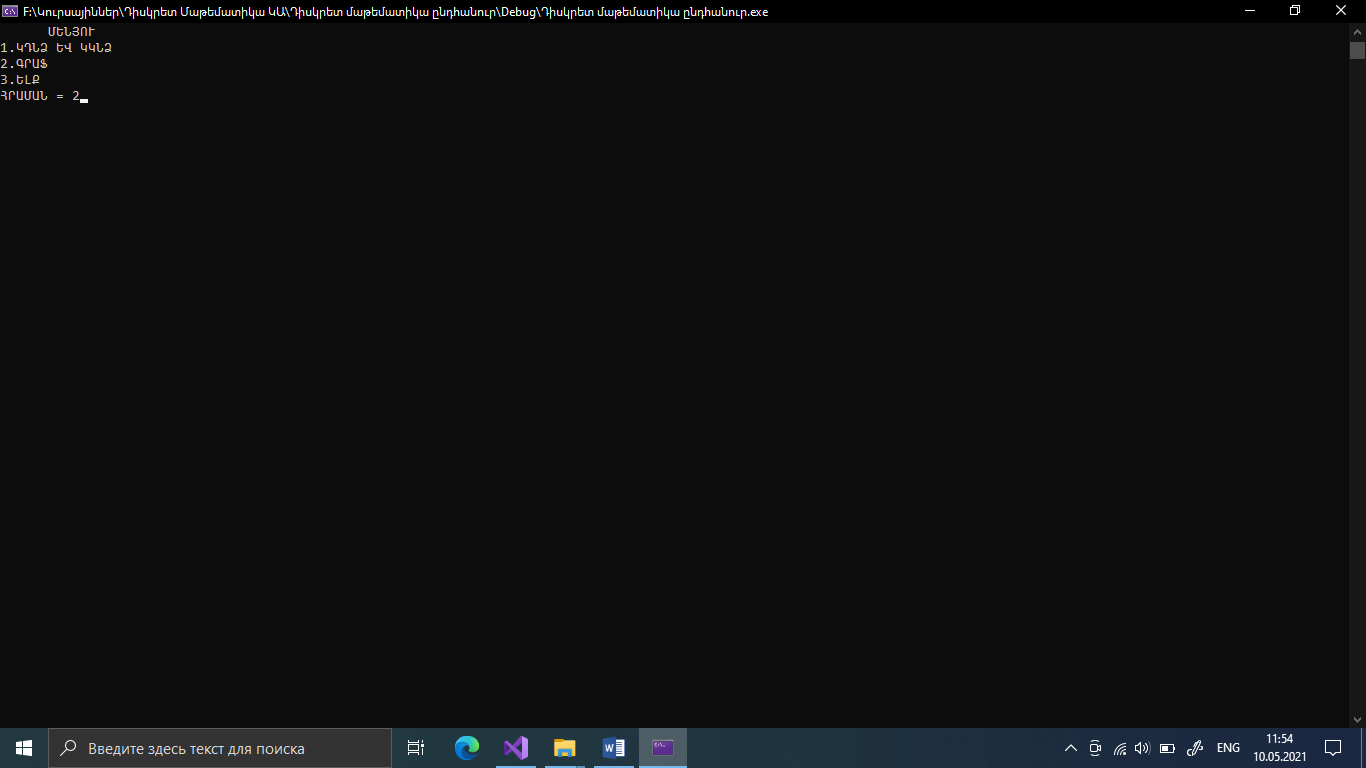
**return** **0**;

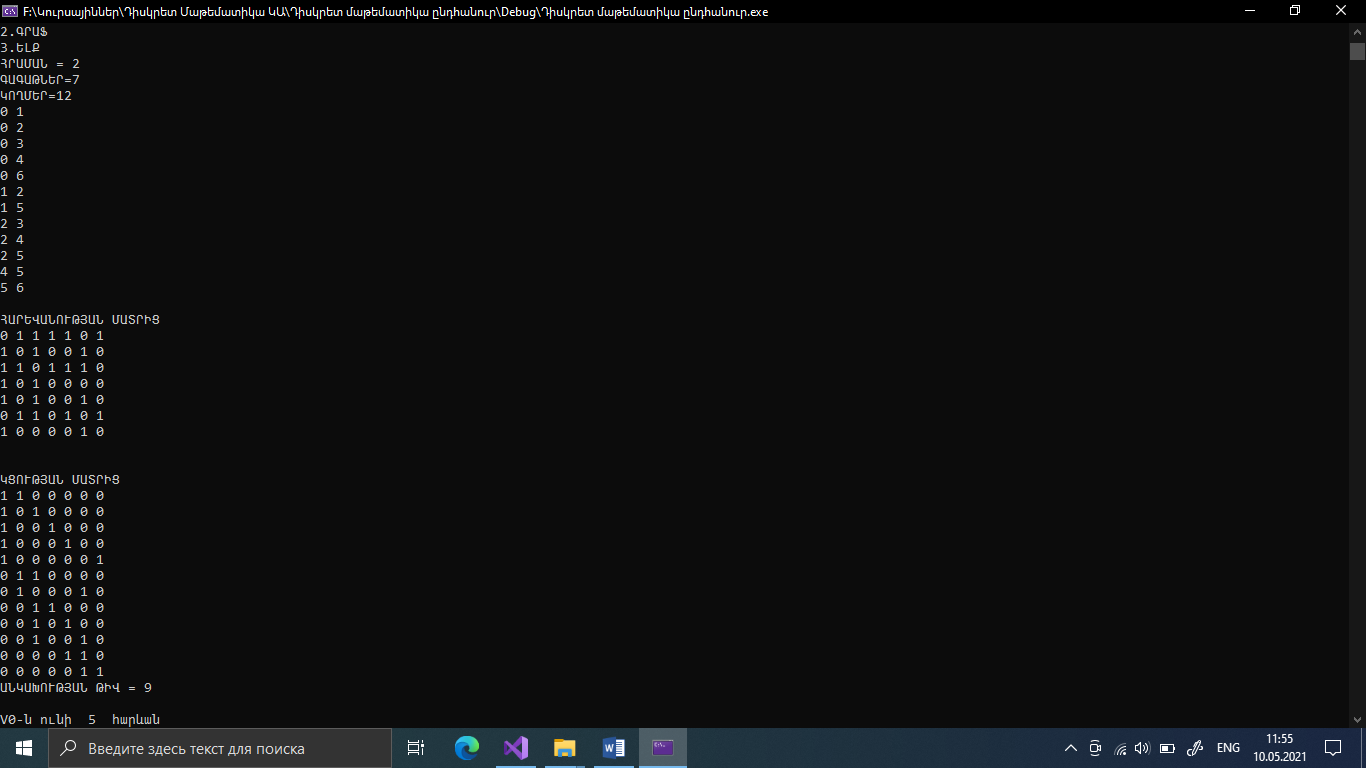
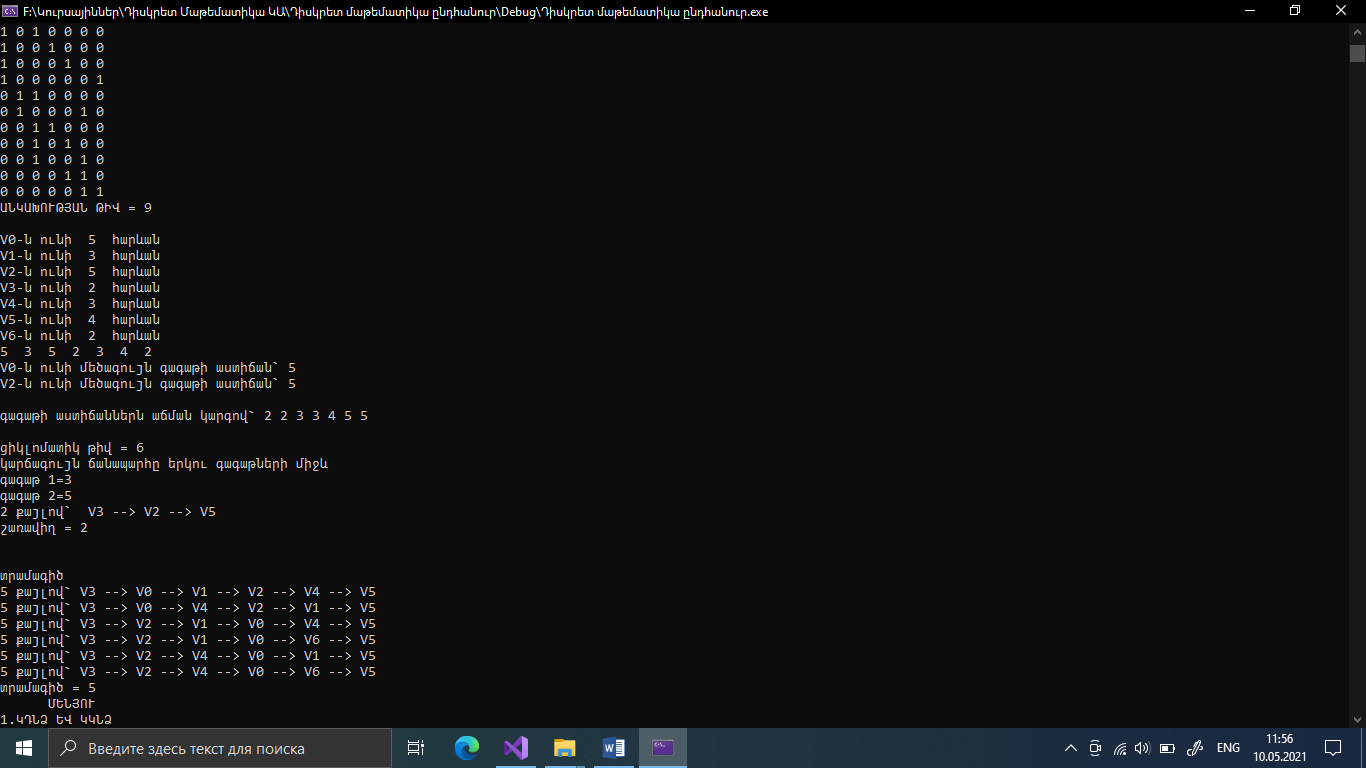
}

Աշխատանքի համար գրվել է ծրագիր C++ ծրագրավորման լեզվով: Այս ծրագիրը հնարավորություն է տալիս կատարել ընտրություն գրաֆների և բուլյան ֆունկցիաների խնդիրների լուծման միջև: Ընտրելով բուլյան ֆունկցիաները, առաջարկվում է ընտրել, թե քանի փոփոխականի ֆունկցիա ենք ուզում ներմուծել, հնարավորություն կա 2,3 կամ 4 փոփոխականի ֆունկցիա ներմուծել: Ընտրելով սրանցից մեկը, պետք է ներմուծել բուլյան ֆունկցիան և ծրագիրը հանում է ֆիկտիվ փոփոխականները (եթե գոյություն ունի), որից հետո արտածում է այդ ֆունկցիայի ԿԿՆՁ-ն և ԿԴՆՁ-ն: Բուլյան ֆունկցիաների հատվածից դուրս գալու համար պետք է ընտրենք ելք կետը: Այժմ հնարավորություն ունենք ընտրելու գրաֆների խնդրի լուծումը: Ընտրելուց հետո պետք է ներմուծենք գրաֆի գագաթների, այնուհետև կողմերի քանակը, որից հետո պետք է ներմուծենք այն գագաթները, որոնք իրար հարևան են: Ներմուծելուց հետո ծրագիրը կտա այդ գրաֆի հարևանության և կցության մատրիցները, ցիկլոմատիկ թիվը, անկախության թիվը, ցույց կտա կախված կամ մեկուսացված գագաթները ինչպես նաև բոլոր գագաթների հարևան գագաթների քանակը, աճման կարգով կդասավորի դրանք: Հնարավորության կա նաև շառավիղը և տրամագիծը գտնելու, դրա համար պետք է ներմուծենք այն երկու գագաթները, որի համար պետք է հաշվել դրանք: Ամբողջ ծրագրի աշխատանքը կարելի է ավարտել՝ ընտրելով մենյուի ելք կետը:









# ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ

Աշխատանքի արդյունքում սովորեցինք դիսկրետ մաթեմատիկայի կիրառման բնագավառները: Հասկացանք, որ մեր սովորած բաժինները՝ բուլյան ֆունկցիաները, կոմբինատորիկան և գրաֆները առօրյա կյանքում մեծ դեր են ունեն:

Բուլյան ֆունկցիաները մեծ կիրառում ունի ՏՏ ոլորտում, այն օգտագործում են միկրոսխեմաների նախագծման մեջ:

Կոմբինատորիկան օգտագործվում է իրական կյանքում տեղի ունեցող պատահարների հավանականության հաշվման համար: Արդեն պարզ է դառնում, որ օգտագործելով կոմբինատորիկան ավելի շատ կխուսափենք սխալներից:

Գրաֆները ուսումնասիրելով հասկացանք, որ այն կիրառվում է բազմաթիվ բնագավառներում, որոնցից ամենապատկերավորը դա որևէ երկրի քաղաքների միջև կապն է:

Աշխատանքի արդյունքում զարգացրեցինք նաև C++ լեզվում մեր ունեցած գիտելիքները, ինչպես նաև մեր ալգորիթմիկ մտածելակերպը:

**ԳՐԱԿԱՆՈՒԹՅՈՒՆ**

1. Դիսկրետ մափեմատիկա, ալգորիթմներ և ծրագրեր: Ն. Ա. Սահակյան
2. Դիսկրետ մաթեմատիկայի դասընթաց: Ռ. Ն. Տոնոյան
3. Ф.А. Новиков: Дискретная Математика для Программистов