

Բովանդակություն

ՆԵՐԱԾՈՒԹՅՈՒՆ	5
ԳԼՈՒԽ 1. ԱՆՀՐԱԺԵՇՏ ՏԵՂԵԿՈՒԹՅՈՒՆՆԵՐ	8
§ 1.1. Գրաֆների վերաբերյալ հիմնական սահմանումներ	8
§ 1.2. Հատումների գրաֆ	10
§ 1.3. Ինտերվալների գրաֆ	12
ԳԼՈՒԽ 2. ԿԱՊՈՒՂՈՒ ՈՒՂԵԳԾՄԱՆ ԽՆԴԻՐԸ	18
§ 2.1. Կապուղու ուղեգծման խնդրի նկարագրությունը	19
§ 2.2. Հիմնական արդյունք	23
§ 2.3. Ձախ ծայրակետի ալգորիթմը	26
§ 2.4. <i>VG</i> -ի ներկման մոտավոր ալգորիթմը	28
ԳԼՈՒԽ 3. ԱԼԳՈՐԻԹՄԻ ԻՐԱԳՈՐԾՈՒՄԸ <i>C++</i> ԾՐԱԳՐԱՎՈՐՄԱՆ ԼԵԶՎՈՎ	29
§ 3.1 Ինտերվալների արտապատկերումը <i>C++</i> դասի միջոցով	29
§ 3.2 Ինտերվալների գրաֆի արտապատկերումը <i>C++</i> դասի միջոցով	32
§ 3.3 Ինտերվալների գրաֆի կառուցումը	34
§ 3.4. Օգտագործման ուղեցույց	41
ԵԶՐԱԿԱՅՈՒԹՅՈՒՆ	44
ՕԳՏԱԳՈՐԾՎԱԾ ԳՐԱԿԱՆՈՒԹՅՈՒՆ	45
ԳԼՈՒԽ 4. ԱՐՏԱԴՐԱՆՔԻ ԻՆՔՆԱՐԺԵՔԻ ԵՎ ՄԻԱՎՈՐ ԳՆԻ ՀԱՇՎԱՐԿ	48
§ 4.1. Հումքի և նյութերի ծախսի որոշումը	49
§ 4.2. Գնովի կիսաֆաբրիկատների և համալրող առարկաների ծախսի հաշվարկ	50
§ 4.3. Էլեկտրաէներգիայի ծախսի հաշվարկ	51
§ 4.4. Արտադրական բանվորների հիմնական աշխատավարձի հաշվարկը	51
§ 4.5. Արտադրական բանվորների լրացուցիչ աշխատավարձի հաշվարկը	53
4.6. Սարքավորումների պահպանման և շահագործման ծախսերի հաշվարկը	53
§ 4.7. Ընդհանուր արտադրական ծախսերի հաշվարկը	54
§ 4.8. Ընդհանուր տնտեսական ծախսերի հաշվարկը	55
§ 4.9. Արտաարտադրական ծախսերի հաշվարկը	55
§ 4.10. Սարքի ընդհանուր ինքնարժքի կալկուլյացիան	55
§ 4.11. Շահույթի և միավորի գնի հաշվարկը	56
ԳԼՈՒԽ 5. <i>K 200-130</i> ԷՆԵՐԳԱԲԼՈՎՆԵՐՈՎ ԿԱՀԱՎՈՐՎԱԾ ՋԷԿ-Ի ԾԽԱԳԱԶԵՐԻ ՀԵՌԱՑՄԱՆ ԾԽՆԵԼՈՒՅԶԻ ԲԱՐՁՐՈՒԹՅԱՆ ՀԱՇՎԱՐԿ	60

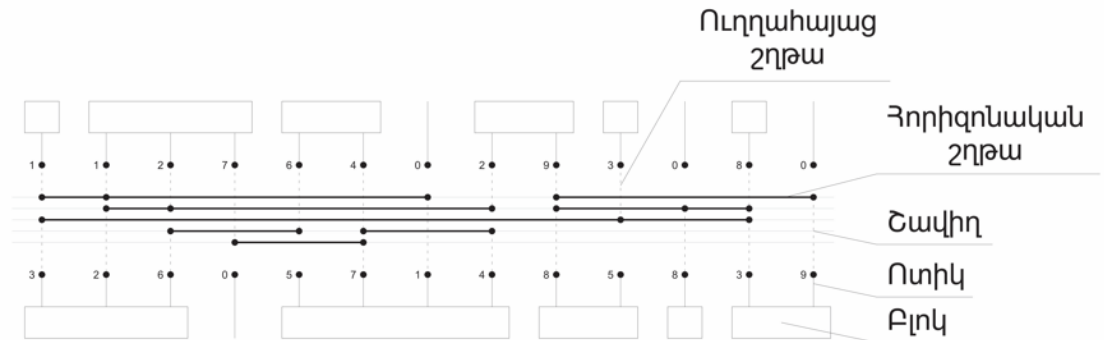
§ 8.1. Ծխնելույզի նվազագույն թույլատրելի բարձրության հաշվարկը	62
§ 8.2. Ծրագրի իրականացումը C++ ծրագրավորման լեզվով	66
ԳԼՈՒԽ 6. ԼԱԶԵՐԱՅԻՆ ՃԱՌԱԳԱՅԹՄԱՆ ՕԳՏԱԳՈՐԾՈՒՄԻՑ ԱՌԱՋԱՑԱԾ ՎՏԱՆԳՆԵՐԻ ԼՈՒՍԱԲԱՆՈՒՄ	69
ՀԱՎԵԼՎԱԾ	73
Թեստավորման արդյունքներ.....	74

ՆԵՐԱԾՈՒԹՅՈՒՆ

Մեծ և գերմեծ ինտեգրալ սխեմաների (ՄԻՍ և ԳՄԻՍ) և տպասալերի ավտոմատ նախագծման գործընթացների դժվար և կարևոր փուլերից է ուղեգծման փուլը: Ինտեգրալ սխեմաներում բազային էլեմենտների և փականների թիվը չափազանց մեծ է՝ այն հասնում է տասնյակ միլիոնների: Այդ պատճառով ինտեգրալ սխեմաների ուղեգծումը իրագործվում է՝ օգտագործելով ուղեգծման բազմաթիվ ալգորիթմներ ընդգրկող ծրագրային փաթեթներ:

Ուղեգծման ալգորիթմների միջոցով որոշվում են բյուրեղի տոպոլոգիական մակերեսի այն ուղիները որոնց վրայով պետք է տարվեն էլեկտրական շղթաներ կրող հաղորդիչներ: Որպես կանոն բյուրեղի մի շերտում տեղադրվում են հորիզոնական, իսկ մյուս շերտում՝ ուղղահայաց շղթաները: Հայտնի է, որ ուղեգծման գործընթացը խլում է ամբողջ նախագծման ժամանակի գրեթե 30%-35%, իսկ բյուրեղի տոպոլոգիական մակերեսի մեծ մասը զբաղեցնում են էլեկտրական շղթաների միացումների հաղորդալարերը: Ուղեգծման գործընթացն արագացնելու համար ուղեգծումը բաժանվում է նախնական (գլոբալ) և վերջնական ենթափուլերի:

Ուղեգծման գործընթացը պարզեցնելու համար ուղեգծման ալգորիթմներն ապավինում են «Բաժանի՛ր և տիրի՛ր» ռազմավարությանը: Նախ կատարվում է նախնական (գլոբալ) ուղեգծում: Քանի որ ուղեգծման ալգորիթմները նախագծում կատարում են մանրամասն միացումներ, նախնական ուղեգծման հիմնական ռազմավարությունը կախված է լինում տեխնոլոգիական սահմանափակումներից: Տարբեր տեխնոլոգիաներ խնդրի լուծման տարբեր մոտեցումներ են ցույց տալիս: Այս փուլի արդյունքում ձևավորվում են շղթայի առանձին մասեր, որոնք կանվանենք բլոկներ (մոդուլներ): Յուրաքանչյուր բլոկ ունի իր մուտքային և ելքային ոտիկները, որոնց միջոցով ապահովվում են կապերն առանձին բլոկների միջև: Բլոկները ինտեգրալ սխեմաների բյուրեղի մակերեսի վրա տեղադրվում են մատրիցային տեսքով, ինչի շնորհիվ նրանց միջև առաջանում են հորիզոնական և ուղղահայաց կապուղիներ: Ուղեգծման հաջորդ՝ վերջնական փուլի խնդիրը այդ կապուղիների միջոցով առանձին բլոկների միջև անհրաժեշտ միացումներն ապահովելն է:



Էլեկտրական շղթաների և դրանց միացությունների չափազանց մեծ լինելու հանգամանքը աննպատակահարմար են դարձնում ալիքային և լաբիրինթոսային տիպի ալգորիթմների և դրանց մոդիֆիկացիաների օգտագործումը, որոնք պահանջում են մեծ հիշողություն և շատ ժամանակ: Ուստի ինտեգրալ սխեմաների ուղեգծման գործընթացում մեծ կարևորություն ունեն կապուղիների ուղեգծման արագագործ ալգորիթմները: Չնայած հայտնի է որ, կապուղիների ուղեգծման խնդիրը ընդհանուր դրվածքով NP-լրիվ է՝ այնուամենայնիվ դրանց ուղեգծման ալգորիթմների մշակման հարցերը հրատապ են և գտնվում են տեսաբանների և ճարտարագետների ուշադրության կենտրոնում: Վերոնշյալ ալգորիթմների կարևորությունը պայմանավորված է նախ նրանով որ այդ ալգորիթմները արագագործ են և պահանջում են քիչ հիշողություն: Բացի այդ կապուղու լայնության փոքրացումը ի վերջո հանգեցնում է բյուրեղի մակերեսի փոքրացմանը, որն էլ իր հերթին փոքրացնում է ինտեգրալ սխեմաների արտադրության խոտանը, նվազեցնում դրանց ինքնարժեքը և բարձրացնում արագագործությունը:

Սույն աշխատանքը նվիրված է մեծ և գերմեծ սխեմաների (ՄԻՍ և ԳՄԻՍ) ավտոմատ նախագծման գործընթացներում կապուղիների ուղեգծման մի նոր մոտավոր ալգորիթմի մշակմանը:

Կապուղին սահմանվում է որպես ուղղանկյուն ցանց ($n \times m$), որն ունի ոտիկների (տերմինալ) 2 տողեր, որոնք ձգվում են ուղղանկյան վերևի և ներքևի կողմերի երկայնքով: Միննույն համարն ունեցող ոտիկները կազմում են մեկ շղթա և պետք է միացվեն իրար

այնպես, որ ոչ մի երկու շղթա չհատվեն: Գրաֆների լեզվով ասած ստացվում է ծառ, որը պարունակում է այդ շղթայի բոլոր ոտիկները:

Յուրաքանչյուր շղթայի համապատասխանեցվում է մեկ ինտերվալ, որի ձախ ծայրակետը որոշվում է շղթայի ամենաձախ ոտիկով, իսկ աջ ծայրակետը՝ ամենաաջ ոտիկով: Այդ ինտերվալը տեղադրվում է հորիզոնական ուղեգծի վրա, իսկ շղթայի ոտիկները այդ ինտերվալի հետ միացվում են մյուս շերտի վրա ուղղահայաց ուղեգծերով: Մի շերտից մյուսը անցումը կատարվում է միջանցիկ անցքերի միջոցով: Հատկանշական է, որ մեկ ինտերվալին կարող են պատկանել երկուսից ավելի ոտիկներ, որոնք նույնպես պետք է ունենան միջանցիկ անցքեր: Հաշվի առնելով այդ փաստը ինտերվալի վրա բացի ծայրակետերից, նշվում են նաև մյուս ոտիկները:

Հայտնի է, որ կապուղու ուղեգծման խնդիրը ընդհանուր դրվածքով NP-լրիվ է: Կապուղու լայնությունն ասելով հասկանում ենք ուղեգծման ժամանակ օգտագործվող հորիզոնական ուղեգծերի քանակը: Գոյություն ունեն կապուղու ուղեգծման խնդրի հնարավոր երկու դեպքեր.

ա) կապուղու լայնությունը ֆիքսված է,

բ) կապուղու լայնությունը ֆիքսված չէ:

Այս աշխատանքում կդիտարկենք այն դեպքը, երբ կապուղու լայնությունը ֆիքսված չէ:

Դասական մոտեցման դեպքում շղթաների միացությունների համար օգտագործվում է ուղղանկյուն ցանցի հորիզոնական և ուղղահայաց հատվածները, և միացությունները իրականացվում են երկու շերտում: Շղթաների հորիզոնական մասերը մեկ շերտում, իսկ ուղղահայաց մասերը մյուս շերտում և նույն շղթայի հորիզոնական և ուղղահայաց մասերը իրար միացվում են միջանցիկ անցքերով:

ԳԼՈՒԽ 1. ԱՆՀՐԱԺԵՇՏ ՏԵՂԵԿՈՒԹՅՈՒՆՆԵՐ

Այս գլխում կձևաթանանք այն բոլոր սահմանումներին ու թեորեմներին, որոնք օգտագործվում են աշխատանքի ընթացքում:

§ 1.1. Գրաֆների վերաբերյալ հիմնական սահմանումներ

- V, E զույգը կանվանենք գրաֆ և կնշանակենք $G = (V, E)$: Ընդ որում՝ V բազմության տարրերը կանվանենք գրաֆի գագաթներ, իսկ E բազմության տարրերը՝ գրաֆի կողեր:
- $G = (V, E)$ գրաֆը կոչվում է լրիվ, եթե նրա կամայական երկու գագաթները հարևան են (կող են կազմում):
- $G = (V, E)$ գրաֆում գագաթների և կողերի $v_1, (v_1, v_2), v_2, (v_2, v_3), \dots, v_{k-1}, (v_{k-1}, v_k), v_k$ (1) հաջորդականությունը, որտեղ $(v_i, v_{i+1}) \in E$ և $i = 1, 2, \dots, k-1$, կոչվում է v_1 և v_k գագաթները միացնող երթուղի:
- Եթե (1) հաջորդականության մեջ $v_i = v_k$, ապա երթուղին կոչվում է փակ:
- Կրկնվող կողեր չպարունակող փակ երթուղին կոչվում է ցիկլ:
- $\bar{G} = (V, E')$ գրաֆը կոչվում է $G = (V, E)$ գրաֆի լրացում, եթե ամեն մի v և v' գագաթների համար ($v \neq v'$) տեղի ունի $(v, v') \in E' \leftrightarrow (v, v') \notin E$:
- $G_1 = (V_1, E_1)$ և $G_2 = (V_2, E_2)$ գրաֆները անվանում են իզոմորֆ և նշանակում են $G_1 \cong G_2$, եթե V_1 և V_2 բազմությունների միջև գոյություն ունի այնպիսի $f: V_1 \rightarrow V_2$ արտապատկերում, որ ցանկացած իրարից տարբեր $v_i, v_j \in V_1$ գագաթների համար տեղի ունի հետևյալ պայմանը. $(v_i, v_j) \in E_1 \leftrightarrow (f(v_i), f(v_j)) \in E_2$:
- $G' = (V', E')$ ենթագրաֆը կոչվում է ծնված, եթե այն պարունակում է $G = (V, E)$ -ի այն բոլոր կողերը, որոնց երկու գագաթները պատկանում են V' -ին:
- Գրաֆի գագաթային ներկում ասելով կհասկանանք նրա գագաթներին գույների այնպիսի վերագրում, որ ոչ մի երկու կից գագաթներ չստանան միևնույն գույնը:
- Գրաֆը կոչվում է k -ներկելի, եթե նրա գագաթները հնարավոր է ներկել k գույներով:

- Նվազագույն k -ն, որի համար G գրաֆը k -ներկելի է, կոչվում է գրաֆի քրոմատիկ թիվ (ներկման թիվ) և նշանակում $\gamma(G)$ -ով:

§ 1.2. Հատումների գրաֆ

Ենթադրենք S -ը բազմություն է, իսկ $F = \{S_1, S_2, \dots, S_n\}$ բազմությունը S -ի որոշակի ոչ դատարկ ենթաբազմությունների ընտանիք է, որոնց միավորումը, հենց ինքը՝ S -ն է: F -ի հատումների գրաֆը նշանակվում է $\Omega(F)$ և սահմանվում, որպես գրաֆ, որի գագաթները հանդիսանում են F բազմությունների ընտանիքի տարրերը՝ $V(\Omega(F)) = F$, իսկ S_i և S_j գագաթները կոդ են կազմում, երբ $i \neq j$ և $S_i \cap S_j \neq \emptyset$:

Այսինքն G գրաֆը S -ի վրա, հանդիսանում է հատումների գրաֆ այն դեպքում, երբ նրանում գոյություն ունի ենթաբազմությունների F ընտանիք, որոնց համար G -ն իզոմորֆ է $V(\Omega(F))$ -ին ($G \cong \Omega(F)$):

Թեորեմ. Ցանկացած գրաֆ հատումների գրաֆ է:

Ապացույց. Ենթադրենք S_i -ն G սովորական գրաֆի v_i գագաթի և նրան կից կողերի համախումբն է ($S_i \stackrel{\text{def}}{=} \{v_i\} \cup E(v_i)$): Ակնհայտ է, որ G -ն իզոմորֆ է $\Omega(F)$ -ին, որտեղ $F = \{S_i\}$:

Ելնելով թեորեմից, կարող ենք սահմանել հատումների թիվը.

Սահմանում. Տրված G գրաֆի հատումների թիվը՝ $\omega(G)$ -ն S -ի տարրերի այն նվազագույն քանակն է, որի դեպքում G -ն հատումների գրաֆ է S -ի վրա:

Հետևանք 1ա. Եթե G գրաֆը կապակցված է և գագաթների քանակը՝ $n \geq 3$, ապա $\omega(G) \leq q$ (կողերի քանակ):

Ապացույց. Այս դեպքում S_i -ներից կարելի է հեռացնել գագաթները, այդպիսով ստանալով, որ $S = X(G)$:

Հետևանք 1բ. Եթե G գրաֆը ունի n_0 մեկուսացված գագաթներ, ապա $\omega(G) \leq q + n_0$:

Թեորեմ. Ենթադրենք G -ն $n > 3$ գագաթանի կապակցված գրաֆ է: Այդ դեպքում հատումների թիվը հավասար է G -ի կողերի քանակին ($\omega(G) = q$), այն և միայն այն դեպքում, երբ G -ն չի պարունակում եռանկյուն:

Ապացույց. Նախ ապացուցենք բավարարությունը:

Բավարարություն. Ըստ հետևանք 1ա-ի հարկավոր է ցույց տալ, որ ցանկացած G կապակցված գրաֆի համար, որն ունի նվազագույնը 4 գագաթ, որոնք եռանկյուն չեն կազմում՝ $\omega(G) \geq q$: Ըստ հատումների թվի սահմանման, S բազմության վրա, որտեղ

$|S| = \omega(G)$ ՝ G -ն իզոմորֆ է $\Omega(F)$ -ին: Ենթադրենք S_i -ն G գրաֆի v_i գագաթի համապատասխան բազմությունն է: Քանի որ, G -ն եռանկյուն չի պարունակում, S -ի ոչ մի տարր չի կարող պատկանել S_i -ի 2-ից ավելի բազմությունների: Այսինքն $S_i \cap S_j \neq \emptyset$ այն և միայն այն դեպքում, երբ $v_i v_j$ -ն G -ի կող է հանդիսանում: Այսպիսով կարող ենք համապատասխանություն մտցնել G -ի կողերի և S -ի այն տարրերի միջև, որոնք պատկանում են ճիշտ երկու S_i բազմությունների: Հետևաբար $\omega(G) = |S| \geq q$, այնպես որ $\omega(G) = q$:

Անհրաժեշտությունն ապացուցելու համար ենթադրենք, որ G -ն պարունակում է եռանկյուն: Ենթադրենք G_1 -ը G -ում եռանկյուն չպարունակող ամենամեծ ենթագրաֆն է: Այդ դեպքում $\omega(G_1) = q_1 = |X(G_1)|$: Ենթադրենք որ $G_1 = \Omega(F)$, որտեղ F -ը q_1 հզորությամբ (բազմության մեջ եզակի էլեմենտների քանակը) S բազմության ենթաբազմությունների ընտանիք է: Ենթադրենք նաև, որ x -ը G գրաֆի կող է, որը չի պատկանում G_1 -ին և նշանակենք $G_2 = G_1 + x$: Քանի որ G_1 -ը եռանկյուն չպարունակող առավելագույն գրաֆ է, ապա G_2 -ը կունենա $u_1 u_2 u_3$ եռանկյուն, որտեղ $x = u_1 u_3$: u_1, u_2, u_3 գագաթներին համապատասխանող S -ի ենթաբազմությունները համապատասխանաբար նշանակենք $S_1 S_2 S_3$: Այժմ եթե u_2 -ը G_1 -ում հարևան է միայն u_1 և u_3 , ապա փոխարինենք S_2 -ը $S_1 \cap S_2$ միակ տարրով և ավելացնենք այդ տարրը S_3 -ին: Հակառակ դեպքում S_3 -ին ավելացնենք $S_1 \cap S_2$ -ից ցանկացած տարր: Երկու դեպքում էլ կստանանք S -ի որոշակի բազմությունների F' ընտանիք, այնպես որ $G_2 = \Omega(F')$: Այսպիսով $\omega(G_2) = q_1$, իսկ $|X(G_2)| = q_1 + 1$: Եթե $G_2 \cong G$ -ին ապա ապացույցի կարիք չկա: Հակառակ դեպքում, երբ $G_2 \neq G$, ապա նշանակենք

$$|X(G)| - |X(G_2)| = q_0$$

Այստեղից հետևություն, որ G -ն $q_1 + q_2$ տարրերով հատումների գրաֆ է: Այնուամենայնիվ $q_1 + q_2 = q - 1$: Հետևաբար $\omega(G_1) < q$ -ն ավարտում է ապացույցը:

Տրված քանակով գագաթներով գրաֆի հատումների թվի հնարավոր լավագույն վերին սահմանը տրվել է Էրդոշի, Գուդմանի և Պոսայի կողմից:

Թեորեմ. Ցանկացած 4 և ավելի գագաթանի ($n \geq 4$) գրաֆի համար հատումների թիվը՝ $\omega(G_1) \leq \left\lceil \frac{n^2(G)}{4} \right\rceil$: Ավելի ճշգրիտ գնահատական կախված $n(G)$ -ից գոյություն չունի:

§ 1.3. Ինտերվալների գրաֆ

Այն առաջին անգամ առաջարկել է ամերիկացի մոլեկուլային կենսաբան Բենզերը: Նա առաջարկել է, որ բակտերիալ քրոմոսոմը, որն իրենից ներկայացնում է գեների շարան կարելի է ներկայացնել, որպես իրական գծի վրա փակ ինտերվալ:

Հայտարարենք անկախ առաջարկել է, որ S_i ինտերվալների ցանկացած վերջավոր F ընտանիք կարող է համապատասխանեցվել մի գրաֆի, որը հատումների գրաֆի առումով կլինի $\Omega(F)$ -ը: Ինտերվալ գրաֆ կանվանենք այն գրաֆը, որը իզոմորֆ է F ինտերվալների ընտանիքի $\Omega(F)$ հատումների գրաֆին:

Հատումների գրաֆի մի հետաքրքիր և առանձնահատուկ դեպք է այն գրաֆը, որի S համակարգի բազմությունները որևէ գծային կարգավորված բազմության ինտերվալներ են: Այս դեպքում հատումների գրաֆին նաև անվանում ենք ինտերվալ գրաֆ:

Գծային կարգավորված M բազմության M' ինտերվալը նկարագրվում է, որպես ոչ դատարկ ենթաբազմություն $M' \subseteq M$, որի համար

$$\forall a, b, c \in M (a, b \in M' \& a < c < b = c \in M'):$$

Սովորական $G = (V, E)$ գրաֆը կոչվում է ինտերվալների կամ Հայոշի գրաֆ, եթե գոյություն ունի այնպիսի գծային կարգավորված M բազմություն և այդ բազմության այնպիսի S համակարգ, որ $G(S)$ -ն իզոմորֆ է G -ին:

Որպեսզի ձևակերպենք G գրաֆի ինտերվալ գրաֆ լինելու պայմանը, նախ վերհիշենք հետևյալ սահմանումները.

Մահմանում. Գրաֆի ցիկլի եռանկյունացնող է կոչվում գրաֆի այն կողը, որը ցիկլի որևէ երկու հաջորդական կողերի հետ կազմում է եռանկյուն:

Մահմանում. Գրաֆը կոչվում է եռանկյունացված, եթե նրա ցանկացած ցիկլ պարունակում է գոնե մեկ եռանկյունացնող:

Թեորեմ 1. Ցանկացած ինտերվալների գրաֆ եռանկյունացված է:

Ապացույց. Ենթադրենք $G = (V, E)$ -ն ինտերվալ գրաֆ է, իսկ C -ն նրա որևէ ցիկլ: Եթե G -ում առկա է C_3 ցիկլ, այսինքն եռանկյուն, ապա ակնհայտ է, որ այդ ցիկլի յուրաքանչյուր կող հանդիսանում է այդ ցիկլի եռանկյունացնող: Այժմ ենթադրենք, ունենք հետևյալ ցիկլը.

$$C = v_0, (v_0, v_1), v_1, (v_1, v_2), \dots, v_{k-1}, (v_{k-1}, v_0), v_0$$

որտեղ $k \geq 4$: Ըստ թեորեմի պայմանի գոյություն ունի այնպիսի գծային կարգավորված M բազմություն և $n \stackrel{\text{def}}{=} n(G)$ -ից այնպիսի S համակարգ, որ $G(S)$ -ն իզոմորֆ է G -ին: Ենթադրենք (որևէ իզոմորֆիզմի համար) C ցիկլի v_i գագաթին համապատասխան ինտերվալը M_i -ն է և $M_{i+k} \stackrel{\text{def}}{=} M_i$ ($i = 0, 1, 2, \dots, k - 1$):

Կարող ենք ասել, որ ըստ $G(S)$ -ի սահմանման

$$\forall i \geq 0 \quad (M_i \cap M_{i+1} \neq \emptyset) \quad (1)$$

Ենթադրենք նաև, որ չնայած թեորեմի պնդմանը՝ C ցիկլը չունի եռանկյունացնող, այսինքն

$$\forall i \geq 0 \quad (M_i \cap M_{i+2} \neq \emptyset) \quad (2)$$

(1) և (2)-ից հետևում է, որ M_0, M_1, \dots, M_{k-1} ինտերվալների փոխդասավորությունը կարող է լինել միայն ինչպես նկ. 1-ում:



(նկար 1)

Այդ դեպքում՝

$$\forall i \geq 2, \forall m \in M_i, \forall m' \in M_0 \quad (m > m'),$$

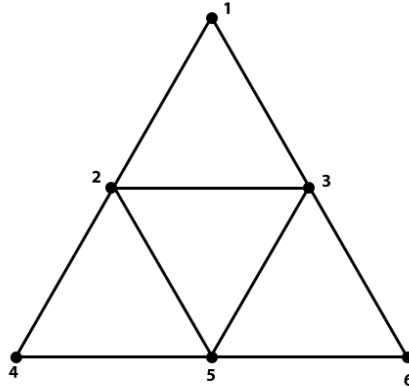
կամ համապատասխանաբար

$$\forall i \geq 2, \forall m \in M_i, \forall m' \in M_0 \quad (m < m'),$$

սակայն երկու դեպքում էլ առաջանում է հակասություն, երբ $i = k$:

Թեորեմն ապացուցված է:

Նկ. 2-ում պարզ երևում է, որ եռանկյունացվածությունը հանդիսանում է գրաֆի՝ ինտերվալներով ներկայացնելի լինելու անհրաժեշտ, բայց ոչ բավարար պայման:

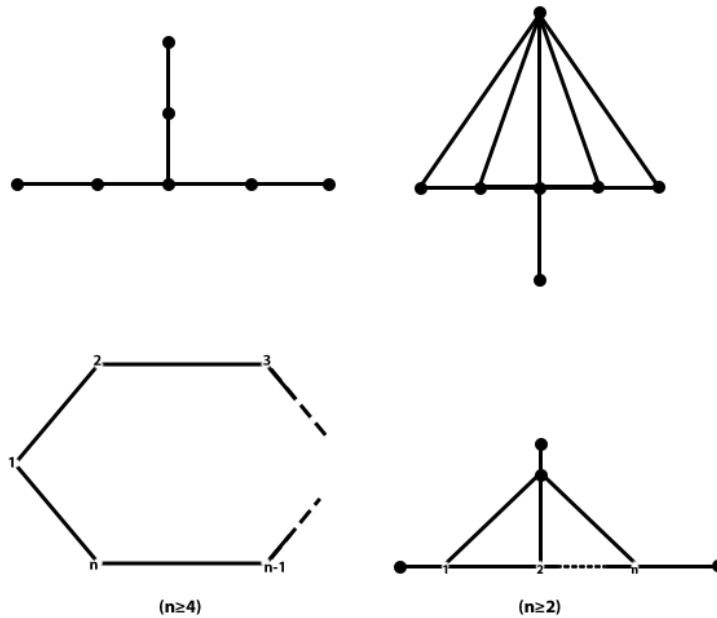


(նկար 2)

Մտածանում. Գրաֆը կոչվում է աստերոիդալ, եթե նրանում առկա v_1, v_2, v_3 գագաթների համար գոյություն ունեն այնպիսի Q_1, Q_2, Q_3 շղթաներ, որ Q_i շղթան միացնում է v_j և v_k գագաթները և այդ շղթայի ոչ մի գագաթ կից չէ v_i -ին ($i, j, k = 1, 2, 3; i \neq j \neq k \neq i$):

Ց. Լեկկերկերն ու Պ. Բոլլանդը 1962 թվականին ապացուցել են, որ սովորական գրաֆը հանդիսանում է ինտերվալ գրաֆ այն և միայն այն դեպքում, երբ այն եռանկյունացված է, բայց աստերոիդալ չէ: Այս պայմանը չնայած թվացյալ բարդությանը բավականին արդյունավետ է գործնական ստուգումների համար: Այն նաև թույլ է տվել հեղինակներին սահմանել ինտերվալ գրաֆների ևս մեկ բնութագիր.

Սովորական գրաֆը հանդիսանում է ինտերվալների գրաֆ այն և միայն այն դեպքում, եթե չի պարունակում նկ. 3-ում պատկերված ենթագրաֆներից որևէ մեկը:



(Նկար 3)

1964 թվականին Հիլմորն ու Հոֆֆմանը առաջարկել են մեկ այլ բնութագիր.

Թեորեմ 2. $G = (V, E)$ սովորական գրաֆը հանդիսանում է ինտերվալների գրաֆ այն և միայն այն դեպքում, երբ նրա ցանկացած C_4 պարզ ցիկլ պարունակում է գոնե մեկ եռանկյունացնող, իսկ G -ի լրացում գրաֆը՝ $\overline{G} = (V, \overline{E})$ տրանզիտիվ ուղղորդելի է:

Ապացույց. Սկզբում ենթադրենք G -ն ինտերվալների գրաֆ է: Ըստ թեորեմ 1-ի G -ում ցանկացած ցիկլ՝ այդ թվում նաև 4 երկարությամբ պարզ ցիկլ պարունակում է եռանկյունացնող: Ուղղորդելով \overline{G} գրաֆի $\overline{xy} \in \overline{E}$ կողը x -ից y , երբ x գագաթի բոլոր տարրերը փոքր են y -ի բոլոր տարրերից, կստանանք \overline{G} գրաֆի տրանզիտիվ կողմնորոշում:

Այժմ ենթադրենք հակառակը՝ ցանկացած 4 երկարությամբ պարզ ցիկլ ունի գոնե մեկ եռանկյունացնող, իսկ $\overline{G} = (V, \overline{E})$ գրաֆը արդեն տրանզիտիվ կողմնորոշված է: $\overline{xy} \in \overline{E}$ -ով կնշանակենք այն կողը, որը միացնում է իրար կից x և y գագաթները, իսկ $\overline{xy} \in \overline{E}$ -ով x գագաթից դեպի y կողմնորոշված կողը: Սկզբում ապացուցենք հետևյալ համակարգի ճշմարտացիությունը.

Ենթադրենք x, y, z, t գազաթներն ընտրված են այնպիսին, որ

$$\begin{cases} \overline{xy} \notin \overline{E} \text{ և } \overline{zt} \notin \overline{E} \text{ և } \overline{xz} \in \overline{E} \text{ և } \overline{yt} \in \overline{E} \\ (\overline{xz} \in \overline{E} \text{ և } \overline{yt} \in \overline{E}) \text{ կամ } (\overline{zx} \in \overline{E} \text{ և } \overline{ty} \in \overline{E}) \end{cases} \quad (*)$$

Քանի որ G -ն սովորական գրաֆ է, համակարգը $x = y$ և $z = t$ արժեքների դեպքում տրիվիալ դեպքն է: $x = y$, $z \neq t$ և $x \neq y$, $z = t$ արժեքների դեպքում համակարգի ճշմարտացիությունը անմիջապես հետևում է \overline{G} գրաֆի տրանզիտիվ կոդմնորոշված լինելուց:

Դիտարկենք $x \neq y$ և $z \neq t$ դեպքը.

$\overline{xt} \in E$ & $\overline{yz} \in E$ դեպքը չի կարող տեղի ունենալ, այլապես G գրաֆի

$$x \overline{xt} t \overline{zt} z \overline{yz} y \overline{xy} x$$

4 երկարության պարզ ցիկլը չի ունենա եռանկյունացնող: Ուրեմն կա՛մ $\overline{xt} \in \overline{E}$, կա՛մ $\overline{yz} \in \overline{E}$: Ենթադրենք $\overline{xt} \in \overline{E}$: Պարզ է, որ այդ դեպքում \overline{xt} կողի ցանկացած կոդմնորոշման դեպքում \overline{G} կոդմնորոշված գրաֆում չի կարող տեղի ունենալ $\overline{xz} \in \overline{E}$ & $\overline{ty} \in \overline{E}$ կամ $\overline{zx} \in \overline{E}$ & $\overline{yt} \in \overline{E}$ պայմաններից և ոչ մեկը, այլապես կխախտվի տրանզիտիվությունը:

Նույն հակասությունը ստացվում է նաև $\overline{yz} \in \overline{E}$ ենթադրության դեպքում: (*) պնդումն ապացուցվեց:

M -ով նշանակենք G գրաֆի բոլոր առավելագույն լրիվ ենթագրաֆների բազմությունը: Այդ ենթագրաֆների համախումբը պարունակում է G գրաֆի բոլոր գազաթներն ու կողերը: M -ում «<» հաջորդականությունը սահմանենք հետևյալ կերպ.

M -ից վերցնենք իրարից տարբեր F և F' տարրերը: Քանի որ G -ի F և F' ենթագրաֆները առավելագույն լրիվ ենթագրաֆներ են, ապա F -ում և F' -ում գոյություն ունեն համապատասխանաբար x և x' գազաթներ այնպես, որ $x \neq x'$ & $\overline{xx'} \notin E$, այսինքն $\overline{xx'} \in \overline{E}$: \overline{G} կոդմնորոշված գրաֆի կողերը, որոնք կապում են F -ի գազաթները F' -ի գազաթներին, ուղղված են կամ F -ից F' , կամ F' -ից F : Առաջին դեպքում կստացվի $F < F'$, երկրորդ դեպքում՝ $F' < F$: M -ից կամայական 2 էլեմենտ կլինեն համեմատելի: Յույց տանք որ M -ում «<» տրանզիտիվ է:

Ենթադրենք հակառակը, որ ներմուծած հարաբերությունը տրանզիտիվ չէ, այլ կերպ ասած գոյություն ունեն այնպիսի $F, F', F'' \in M$, որոնց համար $F < F' < F'' < F$:

Դա նշանակում է որ \bar{G} կողմնորոշված գրաֆում գոյություն ունեն $\overrightarrow{xx'}, \overrightarrow{y'y''}, \overrightarrow{z''z} \in \bar{E}$, որոնք ուղղված են համապատասխանաբար F -ից F' , F' -ից F'' , և F'' -ից F : Անհնար է, որ $x = z$ & $x' = y' & y'' = z''$, այլապես կիսախտվեր \bar{G} կողմնորոշված գրաֆի տրանզիտիվությունը: Ենթադրենք $x \neq z$, ապա $\overrightarrow{y'z} \in \bar{E}$, քանի որ $\overrightarrow{y'z} \in \bar{E}$ դեպքում \bar{G} կողմնորոշված գրաֆի $\overrightarrow{y'y''}$ և $\overrightarrow{z''z}$ կողմնորոշված կողերը կհակասեր (*)-ին y', z, y'', z'' գագաթների համար: Շարունակելով կիրառել (*)-ը x, z, x', y' գագաթներին, կստանանք $\overrightarrow{zy'} \in \bar{E}$: \bar{G} -ի տրանզիտիվությունից հետևում է արդեն, որ $\overrightarrow{z''y'} \in \bar{E}$ և $\overrightarrow{y''z''} \in \bar{E}$, բայց վերջինը հակասում է $\overrightarrow{y''z''} \in E$:

Այսպիսով, M -ի տարրեր էլեմենտներ համեմատելի են և դասավորվածության կարգը տրանզիտիվ է, այդ պատճառով M -ի դասավորությունը գծային է: G գրաֆի յուրաքանչյուր $x \in X$ գագաթին կհամապատասխանեցնենք G գրաֆի $M_x \subseteq M$ այն մաքսիմալ լրիվ ենթագրաֆների բազմությունը, որոնք պարունակում են x : M_x ենթաբազմության յուրաքանչյուր x իրենից ներկայացնում է M -ից ինտերվալ: Իրոք, $F \in M_x, F'' \in M_x$ և $F < F' < F''$, բայց $F' \notin M_x$: Վերջինս նշանակում է որ G գրաֆի x գագաթը չի պատկանում F' : Բայց քանի որ F' -ը մաքսիմալ լրիվ ենթագրաֆ է, նրանում կգտնվի այնպիսի y , որ $\overrightarrow{xy} \notin E$, իսկ դա անհնար է, քանզի \bar{G} կողմնորոշված գրաֆի \overrightarrow{xy} կողը $F < F'$ -ի դեպքում պետք է ուղղված լինի x -ից y , իսկ $F' < F''$ դեպքում y -ից x :

Վերջապես, M -ի M_x ենթաբազմության $S \triangleq \{M_x / x \in X\}$ ինտերվալների համախումբը համապատասխանում է G -ի բոլոր գագաթներին: $L(S)$ հատումների գրաֆը իզոմորֆ է G -ին, այլապես եթե $\overrightarrow{xy} \in E$, ապա $M_x \cap M_y$ հատումը պարունակում է G գրաֆի այն մաքսիմալ լրիվ ենթագրաֆը, որին պատկանում է \overrightarrow{xy} կողը, այսինքն $M_x \cap M_y \neq \emptyset$, իսկ եթե $\overrightarrow{xy} \notin E$, ապա G -ի ոչ մի լրիվ ենթագրաֆ չի կարող միաժամանակ պարունակել x և y գագաթները, որտեղից էլ $M_x \cap M_y = \emptyset$: Թեորեմն ապացուցված է:

ԳԼՈՒԽ 2. ԿԱՊՈՒՂՈՒ ՈՒՂԵԳԾՄԱՆ ԽՆԴԻՐԸ

Ինչպես արդեն նշել ենք մեծ և գերմեծ ինտեգրալ սխեմաների (ՄԻՍ և ԳՄԻՍ) և տպասալերի ավտոմատ նախագծման գործընթացների կարևոր փուլերից է ուղեգծման փուլը: Կապուղու ուղեգծման առաջին ավգործիքն առաջարկվել է Ակիհիրո Հաշիմոտոյի կողմից: Նրանից հետո այն ուսումնասիրվել է բազմաթիվ գիտնականների կողմից, ովքեր նախագծել են խնդրի լուծման տարբեր ավգործիքներ: Ա. Լապոն ապացուցել է, որ սահմանափակումներով կապուղու ուղեգծման խնդիրը NP-լրիվ է: Այս գլխում կապացուցենք, որ ցանկացած k դրական թվի համար գոյություն ունի C սահմանափակումներով կապուղի, որի համար $\tau(C) > \varphi(HG) + L(VG) + k$, որտեղ

$\tau(C)$ -ն կապուղու հաստությունն է,

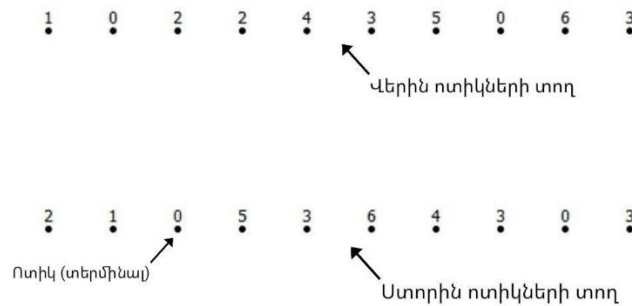
$\varphi(HG)$ - HG հորիզոնական սահմանափակումների գրաֆի առավելագույն ենթագրաֆի զագաթների քանակն է, իսկ

$L(VG)$ - VG ուղղահայաց սահմանափակումների գրաֆում ամենաերկար կողմնորոշված շղթայի երկարությունը:

§ 2.1. Կապուղու ուղեգծման խնդրի նկարագրությունը

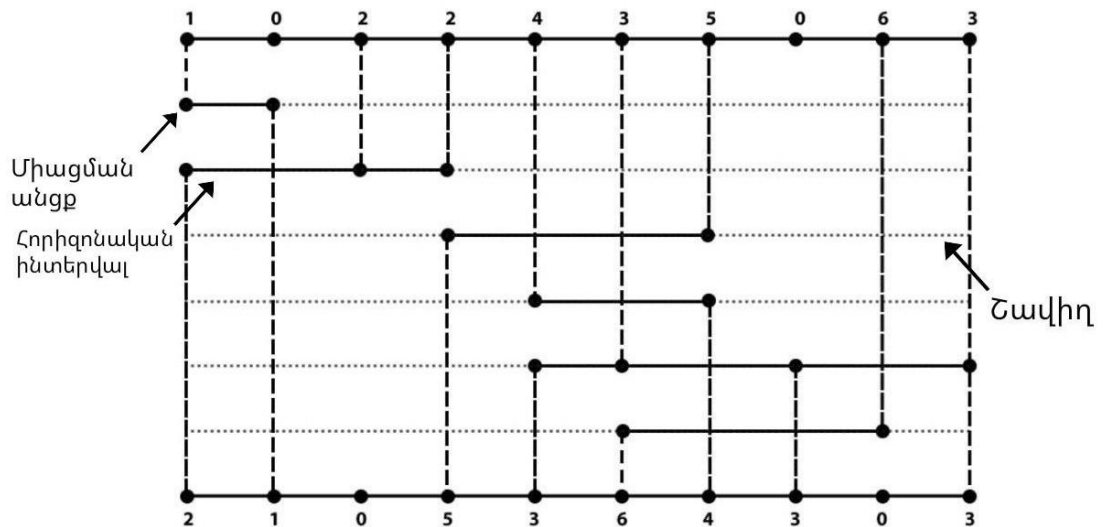
Խնդիրն ավելի մանրամասն նկարագրելու համար, պետք է սահմանենք հետևյալը.

C կապուղին ոչ բացասական կոորդինատներով ոտիկների (տերմինալների) զանգվածների զույգ է. $T = t(1), t(2), \dots, t(n)$ և $B = b(1), b(2), \dots, b(n)$: $n \times m$ ուղղանկյան վերին(t) և ստորին(b) տողերում գտնվող ոտիկները, որոնք համապատասխանաբար ունեն $t(i)$ և $b(i)$ նիշերը՝ պետք է համարակալված լինեն այնպես, որ վերին տողում ցանկացած ոտիկ՝ ստորին տողում ունենա համապատասխան նիշով գոնե մեկ ոտիկ:



(նկար 2.1)

Այսպիսով նույն i թվով համարակալված համապատասխան ոտիկները պետք է իրար միանան i համարի շղթայով: Օ համարով նշանակված ոտիկները կանվանենք ազատ ոտիկներ (տերմինալներ) և նրանց շղթա չենք միացնի: Յուրաքանչյուր i շղթայի հետ համապատասխանեցնենք $I(i)$ ինտերվալը, որտեղ $I(i)$ ձախ(աջ) կետը կլինի նվազագույն(առավելագույն) սյան j համարը, այնպես որ $t(j)$ -ն կամ $b(j)$ -ն հավասար են i -ին: Նախ դիտարկենք սահմանափակումներով ուղեգծման խնդիր, որտեղ ցանկացած շղթայի համար նախատեսված է մեկ հորիզոնական շավիղ: Այս դեպքում խնդրի լուծումը շատ պարզ է: Յուրաքանչյուր շղթա կներկայացնենք, որպես առանձին հատված, որոնք մյուս շերտում իրենց ոտիկների հետ միանում են ուղղահայաց ճյուղերով (տես նկ. 2.2):



(Նկար 2.2)

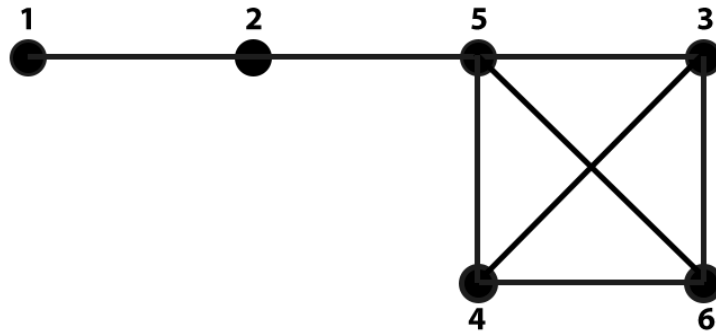
Այսպիսի լուծումը իհարկե կատարում է բոլոր անհրաժեշտ միացումները, սակայն այն ամեն շղթայի համար օգտագործում է մեկ շավիղ, ինչը նպատակահարմար չէ: Ավելի լավ արդյունքի կարող ենք հասնել՝ իրար հետ չհատվող հորիզոնական ինտերվալները միևնույն տողի վրա տեղադրելով: Սակայն այս դեպքում պետք է հաշվի առնենք նաև ուղղահայաց սահմանափակումները: Այսպիսով սահմանափակումներով ուղեգծման խնդրի նպատակը նվազագույն հորիզոնական շավիղների քանակը որոշելն է: Այս խնդրի լուծման ալգորիթմի առավելությունն իր պարզության և նվազագույն քանակի շավիղների օգտագործման մեջ է: Երկշերտ ուղեգծման ժամանակ առաջին շերտի վրա տեղադրվում են բոլոր հորիզոնական շղթաները, իսկ ուղղահայաց շղթաները տեղադրվում են երկրորդ շերտի վրա:

Եթե երկու հորիզոնական շղթաներ չեն հատվում իրար հետ, ապա նրանք կարող են տեղակայվել միևնույն հորիզոնական շավիղի վրա, հակառակ դեպքում ստիպված ենք նրանց տեղակայել տարբեր շավիղներում:

Ուղղահայաց շավիղներում շղթաները նույնպես պետք է չհատվեն: Ակնհայտ է, որ ավելի քարձր հորիզոնական շավիղում գտնվող շղթայի ինտերվալը պետք է տեղադրված լինի ավելի վերև, քան ավելի ներքևում գտնվողինը: Հետևաբար տեղի ունեն նաև սահմանափակումներ ուղղահայաց շղթաների միջև:

Այսպիսով ցանկացած ուղեգծման խնդիր կհամապատասխանեցնենք երկու սահմանափակումների գրաֆների հետ՝ հորիզոնական և ուղղահայաց: Երկու գրաֆներում էլ շղթաները կներկայացնենք, որպես գագաթներ:

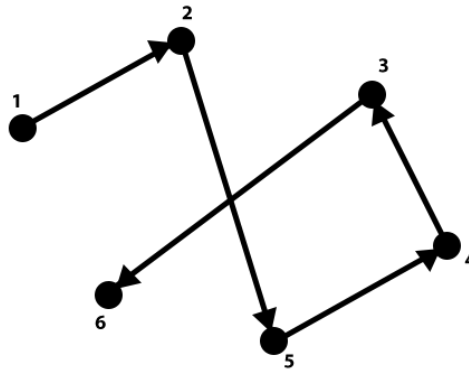
Հորիզոնական սահմանափակումների $HG = (V, E)$ գրաֆը, չկողմնորոշված գրաֆ է, որտեղ $I \in V$ գագաթը ներկայացնում է i համարով շղթան, իսկ $(i, j) \in E$ եթե i և j հորիզոնական ինտերվալները հատվում են: Այսպիսով այն հետևյալ տեսքի ինտերվալ գրաֆ է.



(Նկար 2.3)

Ուղղահայաց սահմանափակումների $VG = (V, E')$ գրաֆը կողմնորոշված գրաֆ է, որի յուրաքանչյուր $I \in V$ գագաթը համապատասխանում է i համարով շղթային և յուրաքանչյուր j սյուն, որի համար $t(j)$ -ն և $b(j)$ -ն ոչ զրոյական թվեր են, առաջացնում է $t(j)$ -ից $b(j)$ կողմնորոշված կող: Այսինքն, եթե ցանկացած երկու շղթա ունեն ոտիկներ նույն սյան երկու ծայրերում, ապա $VG = (V, E')$ գրաֆում նրանց միջև կտանենք կողմնորոշված կող՝ վերևի ոտիկին համապատասխանող գագաթից դեպի ներքևի ոտիկին համապատասխանող գագաթ: Եթե VG գրաֆում գոյություն ունի կողմնորոշված ցիկլ, ապա ուղեգծման պահանջները չեն կարող բավարարվել, առանց այդ շղթաները մասնատելու: Երբեմն նույնիսկ շղթաները մասնատելով հնարավոր չէ հասնել ցանկալի արդյունքին: Օրինակ՝ $T = (1, 2)$ և $B = (2, 1)$ -ը հնարավոր չէ ուղեգծել:

Հեշտ է տեսնել, որ հորիզոնական սահմանափակումների գրաֆը իր մեջ պարունակում է ուղղահայաց սահմանափակումների գրաֆին: Նկ. 2.4-ում պատկերված է նկ. 2.1-ում պատկերված օրինակի ուղղահայաց սահմանափակումների գրաֆը:



(Նկար 2.4)

Այս երկու գրաֆները մեզ հնարավորություն են տալիս ուղեգծման խնդիրը դիտարկել որպես գրաֆների տեսական խնդիր:

Նախագծողի նպատակը նվազագույն շավիղներ օգտագործելով կապողու երթուղավորումն է: C կապուղու $\tau(C)$ հաստությունը (սյունների քանակը) հաստատուն է և որոշված նախագծողի կողմից: Այսպիսով մենք կարող ենք ավելացնել հորիզոնական շավիղներ, բայց սյունների քանակը փոխել չենք կարող:

Կապուղու հաստության որոշման հարցում կարևոր դեր է խաղում հորիզոնական սահմանափակումների HG գրաֆը: HG գրաֆի առավելագույն լրիվ ենթագրաֆի գագաթների քանակը $\varphi(HG)$ ՝ նրա զույգ առ զույգ հատվող ինտերվալների առավելագույն թիվն է: Քանի որ 2 շղթաներ, որոնց ինտերվալները հատվում են չեն կարող լինել նույն հորիզոնական շավիղում, ապա HG գրաֆի առավելագույն լրիվ ենթագրաֆի գագաթների քանակը ($\varphi(HG)$) իրենից ներկայացնում է կապուղու հաստության ստորին գնահատականը. $\tau(C) \geq \varphi(HG)$: Այս սահմանը շատ կարևոր է և հաճախ առկա միակ ստորին սահմանն է:

Ուսումնասիրենք VG ուղղահայաց սահմանափակումների գրաֆի ազդեցությունը կապուղու հաստության վրա: VG գրաֆի ամենաերկար կողմնորոշված շղթայի $L(VG)$ երկարությունը իրենից ներկայացնում է կապուղու հաստության մեկ այլ ստորին գնահատական. $\tau(C) \geq L(VG)$: Այս փաստը հետևում է այն բանից, որ երկու շղթաներ որոնք գտնվում են նույն կողմնորոշված շղթայի վրա չեն կարող տեղակայվել նույն ուղղահայաց մագիստրալում:

§ 2.2. Հիմնական արդյունք

Ակնհայտ է որ կապուղու հաստության “լավ” ստորին և վերին գնահատականները կարևոր են, քանի որ դրանք կարող են գնահատել ինտեգրալ սխեմայի մակերեսը և տեղադրման որակը: Այդ պատճառով տեղին կլինի հարցը այն մասին, թե արդյոք հնարավոր է $\varphi(HG)$ և $L(VG)$ թվերով ստանալ կապուղու հաստության վերին գնահատականը: Ցավոք սահմանափակումներով ուղեգծման համար այն հնարավոր չէ, քանի որ ինչպես կներկայացվի ներքևում կապուղու հաստության որոշման հարցում $\varphi(HG)$ -ն և $L(VG)$ -ն միակ գործոնները չեն:

Հիշենք, որ միևնույն հորիզոնական շավիղի վրա գտնվող շղթաները չեն հատվում: Սա նշանակում է, որ շղթաների ինտերվալների շավիղներին վերագրման գործընթացը կարելի է ներկայացնել HG սահմանափակումների գրաֆի ներկման խնդրի տեսքով: Քանի որ HG -ն ինտերվալ գրաֆ է, նրա գագաթները ներկելու համար կարող ենք օգտագործել հայտնի ձախ ծայրի ալգորիթմը:

Ինչպես արդեն նշել ենք, եթե ուղղահայաց սահմանափակումների VG գրաֆը պարունակում է կողմնորոշված ցիկլ, ապա խնդիրն առանց շղթաների մասնատման անլուծելի է: Այնուամենայնիվ եթե այն չի պարունակում կողմնորոշված ցիկլեր, այդ դեպքում նույնպես ուղեգծման խնդիրը հեռու է պարզ լուծում ունենալուց: Ա. Լապոն (A. LaPaugh) ապացուցել է որ սահմանափակումներով ուղեգծման խնդիրը NP-լրիվ (non-deterministic polynomial-time) է: Հեշտ է տեսնել, որ այդ խնդիրը կարող է ներկայացվել, որպես ուղղահայաց սահմանափակումների VG գրաֆի ներկման խնդիր:

$VG = (V, E')$ գրաֆի k -ներկումը դա հետևյալ “պիտակավորում” է. $f: V \rightarrow \{1, 2, \dots, k\}$: Այստեղ պիտակները հենց գույներն են: k -ներկումը համարվում է ստույգ, եթե ցանկացած $(i, j) \in E$ -ի համար $f(i) \neq f(j)$ և $(k, m) \in E'$ -ի համար $f(k) < f(m)$: VG գրաֆը կոչվում է k -ներկելի, եթե այն ունի պատշաճ k -ներկում: Նվազագույն k թիվը, որի դեպքում VG գրաֆը ստույգ k -ներկելի է՝ կոչվում է քրոմատիկ թիվ և նշանակվում $\chi'(VG)$:

Ըստ Ա. Ֆրանկի ցանկացած կապուղու համար

$$\tau(G) > L(VG) + \varphi(HG) + 2,$$

որտեղ $L(VG)$ -ն VG -ում ամենաերկար կողմնորոշված ճանապարհի երկարությունն է, իսկ $\varphi(HG)$ -ն HG գրաֆի առավելագույն լրիվ ենթագրաֆի գագաթների քանակն է:

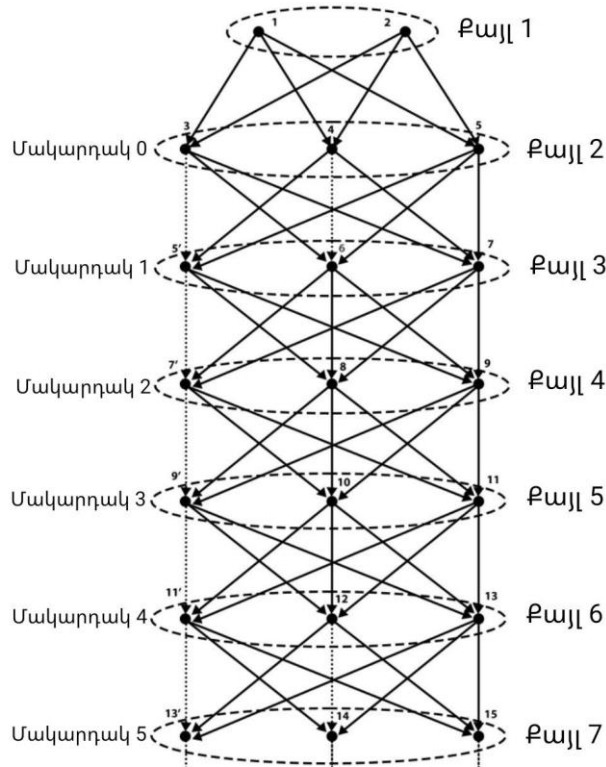
Թեորեմ. Ցանկացած k դրական թվի համար գոյություն ունի C սահմանափակումներով կապուղի, որի համար $\tau(C) > L(VG) + \varphi(HG) + k$:

Ապացույց. Վերցնենք սահմանափակումներով կապուղի, որի սկզբնական վիճակը տրված է նկար 2.5-ում: Նկար 2.6-ում պատկերված է այդ կապուղու VG ուղղահայաց սահմանափակումների գրաֆը: Երթուղին կառուցվում է քայլ առ քայլ: Ամեն քայլում ավելացված գագաթները նկար 2.6-ում ընդգծված են կետագծերով:

2 0 2 2 2 1 1 5 5 4 4 4 3 3 3 7 7 6 6 6 5' 5' 5' 9 9 8 8 8 7' 7' 7' 11 11 10 10 10 9' 9' 9' 13 13 12 12 12 11' 11' 11'

0 1 5 4 3 5 4 3 7 6 7 6 5' 7 6 5' 9 8 9 8 7' 9 8 7' 11 10 11 10 9' 11 10 9' 13 12 13 12 11' 13 12 11' 15 14 15 14 13' 15 14 13'

(Նկար 2.5)



(Նկար 2.6)

G_j -ով նշանակենք VG -ի այն ենթագրաֆը, որը կազմված է $j = 0, 1, \dots$ մակարդակներում գտնվող բոլոր գագաթներից: Թեորեմն ապացուցելու համար կատարենք հետևյալ պնդումները.

1. $\varphi(HG) = 5$
2. $L(G_0) = 1$ և $L(G_{j+1}) > L(G_j) + 1, j = 0, 1, 2, \dots$
3. $\chi'(G_0) = 5$ և $\chi'(G_{j+1}) > \chi'(G_j) + 2, j = 0, 1, 2, \dots$

Ըստ *կասյուղու* կառուցվածքի նույն քայլի գագաթները HG հորիզոնական սահմանափակումների գրաֆում զույգ առ զույգ հարևան են: Կոդմնորոշված կողերը միացնում են միայն հաջորդական քայլերի գագաթներ: Ավելին VG ուղղահայաց սահմանափակումների գրաֆը պարունակում է այդ բոլոր կողերը՝ բացառությամբ $(5,5'), (7,7'), (9,9'), (11,11')$ և այլն: Քանի որ հորիզոնական սահմանափակումները պարունակում են նաև ուղղահայաց սահմանափակումներին, հետևաբար VG -ին պատկանող բոլոր կոդմնորոշված կողերը պատկանում են նաև HG -ին ոչ կոդմնորոշված տեսքով: Դժվար չէ նկատել, որ ամեն հաջորդ քայլ համապատասխան ենթագրաֆում ամենաերկար ճանապարհի երկարությունն ավելացնում է 1-ով: Սա ապացուցում է 1, 2 կետերը:

3-րդ կետն ապացուցելու համար նկատենք, որ եթե գոյություն ունի i և j գագաթները միացնող կոդմնորոշված ճանապարհ, ապա VG -ի ներկման ժամանակ, i գագաթի գույնը պետք է փոքր լինի j գագաթի գույնից: Նկատենք նաև, որ եթե ունենք G_i գրաֆի օպտիմալ ներկում, կարող ենք ստանալ G_{i+1} -ի օպտիմալ ներկումը՝ շարունակելով G_i -ի ներկմանը ավելացնելով ևս 2 գույն: Քանի որ G_0 -ն HG -ում լրիվ ենթագրաֆ է, ապա $\chi'(G_0) = 5$: G_0 -ի յուրաքանչյուր գագաթ՝ բացառությամբ 5-ի, կից է 3-րդ քայլի յուրաքանչյուր գագաթին կոդմնորոշված կողով, հետևաբար 6 և 7 գագաթների գույները պետք է ավելի մեծ լինեն քան G_0 -ի բոլոր գագաթների գույները: Քանի որ 5 և 5' գագաթները միացված չեն կոդմնորոշված կողով, մենք կարող ենք ներկել նրանց նույն գույնով: Այսպիսով G_1 -ի օպտիմալ ներկումը ստացվեց G_0 -ի օպտիմալ ներկմանը 2 գույն ավելացնելով:

Նույն օրինաչափությամբ G_2 -ի օպտիմալ ներկումը կստացվի G_1 -ի օպտիմալ ներկմանն ավելացնելով ևս 2 գույն և այդպես շարունակ: 3-րդ պնդումն ապացուցված է: Ստացանք, որ

ամեն նոր՝ i -րդ քայլի համար ($i > 2$) առավելագույն լրիվ ենթագրաֆի գագաթների քանակը մնում է անփոփոխ, ամենաերկար ճանապարհը ավելանում է 1-ով, իսկ քրոմատիկ թիվը մեծանում է 2-ով: Նկար 2.6-ում գտնվող գրաֆի համար $\varphi(HG) = 5$, $L(VG) = 6$ և $\chi'(VG) = 15$: Ցանկացած k -ի համար կատարելով բավարար քանակությամբ քայլեր կարող ենք կառուցել սահմանափակումներով կապուղի, որի համար $\tau(C) > L(VG) + \varphi(HG) + k$: Թեորեմն ապացուցվեց:

Նշենք, որ վերոնշյալ ենթադրությունը մնում է բաց առանց սահմանափակումների կապուղիների համար:

§ 2.3. Ձախ ծայրակետի ալգորիթմը

Դիցուք տրված է $F = \{f_1, f_2, \dots, f_n\}$ ինտերվալների ընտանիքը որտեղ $f_i = [a_i, b_i]$, a_i -ն կոչվում է f_i ինտերվալի ձախ ծայրակետ, իսկ b_i -ն աջ ծայրակետ: F -ի ներկում ասելով կհասկանանք գույների այնպիսի վերագրում, որ հատվող ինտերվալներին վերագրված գույները լինեն տարբեր: Որպես գույներ կօգտագործենք բնական թվերը՝ $1, 2, \dots, m$:

Ձախ ծայրակետի ալգորիթմն իրագործելու համար առաջին քայլով վերցնենք S_i դատարկ բազմությունը, որին կավելացնենք այն տարրերը, որոնք պետք է ներկվեն i գույնով: Քանի որ որպես գույներ օգտագործում ենք բնական թվերը, առաջին քայլում i գույնը կվերցնենք 1:

Քայլ 0: $i = 1, S_i = \emptyset$,

Քայլ 1: $F' := F$:

Քայլ 2: Հաջորդ քայլով F' -ից ընտրենք այն $f_{i_1} = [a_i, b_i]$ ինտերվալը, որի ձախ ծայրակետը ամենաձախն է՝ $a_{i_1} = \min_{i \in \{1, 2, \dots, n(F')\}} \{a_i\}$: Գտնված f_{i_1} ինտերվալը ավելացնենք i գույնով ներկված ինտերվալների S_i բազմությանը.

$$S_i := S_i \cup \{f_{i_1}\},$$

պարզ լեզվով ասած այն ներկենք i գույնով: Քանի որ ինտերվալն արդեն ներկվել է հեռացնենք այն F' բազմությունից.

$$F' := F' \setminus \{f_{i_1}\}$$

Քայլ 3: F' -ից հեռացնենք այն բոլոր ինտերվալները, որոնք հատվում են f_{i_1} -ի հետ:

Քայլ 4: Եթե $F' \neq \emptyset$, կրկնենք «Քայլ 2»-ը, հակառակ դեպքում՝ անցնենք «Քայլ 5»-ին:

Եթե «Քայլ 4»-ից հետո F' բազմությունը դատարկ է, ապա նշանակում է, որ այլևս չկա այնպիսի ինտերվալ, որը կարելի է ներկել i գույնով և չի պատկանում S_i -ին: Այդ դեպքում F -ից հեռացնենք S_i բազմության տարրերը և եթե $F \neq \emptyset$ կվերադառնանք «Քայլ 1»-ին:

Քայլ 5: $F := F \setminus S_i$: Եթե $F \neq \emptyset$ ապա $i := i + 1$, $S_i = \emptyset$ և վերադառնում ենք «Քայլ 1»:
Հակառակ դեպքում բոլոր ինտերվալները ներկված են՝ ալգորիթմն ավարտված է:

Ենթադրենք k -ն ամենամեծ բնական թիվն է, որ $S_k \neq \emptyset$: Նոր նշանակումներ չմտցնելու համար և պարզության համար համարենք որ f_i -ն S_i -ի ($i = 1, 2, \dots, n$) առաջին ինտերվալն է, որը հատվում է f_k -ի հետ: Պարզ է, որ $\forall j$ -ի համար $j = 1, 2, \dots, k - 1$ գոյություն ունի $f_j \in S_j$, որը հատվում է f_k -ի հետ: Նշանակենք $a = \max_{j \in \{1, \dots, k\}} a_j$ և $b = \min_{j \in \{1, \dots, k\}} b_j$: Դժվար չէ համոզվել որ $[a, b]$ -ի ամեն մի կետ պատկանում է բոլոր f_1, f_2, \dots, f_k ինտերվալներին, հետևաբար եթե $F = \{f_1, f_2, \dots, f_n\}$ ինտերվալների ընտանիքի ներկման համար օգտագործվել են k գույներ, ապա F -ում գոյություն կունենա k հատ գույգ առ գույգ իրար հետ հատվող ինտերվալներ: Ուստի ձախ ծայրակետի ալգորիթմը օպտիմալ է:

§ 2.4. VG -ի ներկման մոտավոր ալգորիթմը

Ալգորիթմն իրագործելու համար ամեն i -րդ քայլում պետք է VG -ում ընտրենք այն բոլոր գագաթները որոնց համար մտնող կողմնորոշված կողերը նվազագույնն են և դրանց վրա կիրառենք ձախ ծայրակետի ալգորիթմը՝ ներկելով նրանց $k = j_{i-1}, j_{i-1} + 1, \dots, j_i$ գույներով:

Քանի որ առաջին քայլի ($i = 1$) առաջին գույնը 1-ն է նշանակենք $j_{i-1} = j_0 = 1$:

Քայլ 0: $i = 1, j_0 = 1$

Քայլ 1: VG -ում ընտրենք այն բոլոր գագաթները որոնց համար մտնող կողմնորոշված կողերը նվազագույնն են: Այդ գագաթների բազմությունը նշանակենք V_i -ով: V_i -ի գագաթներին համապատասխանող ինտերվալների բազմությունը նշանակենք F_i -ով: F_i -ի վրա կիրառենք ձախ ծայրակետի ալգորիթմը: Ենթադրենք F_i -ի ներկման համար օգտագործվեց $k = j_{i-1}, j_{i-1} + 1, \dots, j_i$ գույներ: Դիտարկենք $VG := VG \setminus V_1$ -ը:

VG բազմությունից հեռացնենք V_1 -ը, եթե բազմությունը դատարկ չէ կրկնենք «Քայլ 1»-ը, հակառակ դեպքում ալգորիթմն ավարտված է:

Քայլ 2: $VG := VG \setminus V_1$: Եթե $VG \neq \emptyset$, ապա $i := i + 1$ և կրկնենք «Քայլ 1»-ը: Հակառակ դեպքում բոլոր ինտերվալները ներկված են՝ ալգորիթմն ավարտված է:

ԳԼՈՒԽ 3. ԱԼԳՈՐԻԹՄԻ ԻՐԱԳՈՐԾՈՒՄԸ C++ ԾՐԱԳՐԱՎՈՐՄԱՆ ԼԵԶՎՈՎ

Խնդրի լուծման ալգորիթն իրականացվել է C++ ծրագրավորման լեզվի միջոցով: Գրաֆիկական միջավայրը կառուցվել է Qt ծրագրային լրացման միջոցով: Աշխատանքի շրջանակներում կքննարկենք միայն ինտերվալ գրաֆի ստեղծումն ու ինտերվալների տեղադրումը համապատասխան տողերում: Ստացված տեղեկության գրաֆիկական միջավայրում ներկայացումը չենք քննարկի, սակայն այն հասանելի է աշխատանքին կից տրամադրված ծրագրում:

Սկզբում ծանոթանանք այն կառուցվածքների հետ, որոնք օգտագործվում են ալգորիթմն իրականացնելիս, որից հետո կքննարկենք բուն ալգորիթմը:

§ 3.1 Ինտերվալների արտապատկերումը C++ դասի միջոցով

Ինտերվալները ներկայացված են “IntervalData” դասի (class) միջոցով: Այս դասն իր մեջ պարունակում է ինտերվալին պատկանող գազաթների ինդեքսները, *VG* գրաֆում ինտերվալին համապատասխան գազաթից դուրս եկող կողերով կից գազաթների համարները, ինտերվալների գրաֆում իր ինդեքսը, *VG* գրաֆում դեպի իրեն համապատասխանող գազաթը տանող ամենաներկար ճանապարհի երկարությունը:

```
class IntervalData
{
public:
    . . . .
    friend bool operator==( const IntervalData& interval1, const IntervalData& interval2 );
    friend bool operator<( const IntervalData& interval1, const IntervalData& interval2 );
    friend bool operator>( const IntervalData& interval1, const IntervalData& interval2 );
    friend bool operator<<( const IntervalData& interval1, const IntervalData& interval2 );
    . . . .
protected:
```

```

std::set< size_t > m_setCols;
std::set< size_t > m_setOutConnections;
size_t m_nIndex;
size_t m_nVertexValue;
size_t m_nRank;
};

```

Ինտերվալին պատկանող ոտիկների ինդեքսները պահվում են կարգավորված `m_setCols` զանգվածով: Կարգավորված զանգվածը (`std::set`) մեզ հնարավորություն է տալիս հեշտությամբ ստանալ ինտերվալի ձախ և աջ ծայրակետերը դիմելով զանգվածի համապատասխանաբար առաջին և վերջին էլեմենտներին:

VG ուղղահայաց սահմանափակումների գրաֆում ինտերվալին համապատասխան գագաթից դուրս եկող կողերով կից գագաթների համարները կպահենք `m_setOutConnections` կարգավորված բազմության մեջ: Նույն գրաֆում դեպի գագաթը տանող ամենաերկար ճանապարհի երկարությունը կպահենք `m_nRank` փոփոխականով: Ինտերվալների գրաֆում սեփական տարրի ինդեքսը կպահենք `m_nIndex` փոփոխականի միջոցով, իսկ ոտիկի (գագաթ) նիշը՝ `m_nVertexValue`-ի:

Ինտերվալների գրաֆը կազմելու համար հարկավոր է սահմանել երկու ինտերվալների միջև համեմատության գործողություններ: Երկու ինտերվալների միջև սահմանենք մեծ (`>`), փոքր (`<`), բացարձակ մեծ (`>>`) և բացարձակ փոքր (`<<`) համեմատության գործողությունները.

Կասենք, որ առաջին ինտերվալը փոքր է երկրորդից, եթե նրա ձախ ծայրակետի ինդեքսը փոքր է երկրորդ ինտերվալի ձախ ծայրակետի ինդեքսից:

```

bool operator<( const IntervalData& interval1, const IntervalData& interval2 )
{
    size_t nStart1 = *( interval1.m_setCols.begin() ),
           nStart2 = *( interval2.m_setCols.begin() );

    return nStart1 < nStart2;
}

```

Կասենք առաջին ինտերվալը մեծ է երկրորդից, եթե նրա ձախ ծայրակետի ինդեքսը մեծ է երկրորդ ինտերվալի ձախ ծայրակետի ինդեքսից: Այսինքն `>>` գործողությունը `<<` գործողության մաքուր հակադարձն է:

```
bool operator>( const IntervalData& interval1, const IntervalData& interval2 )
{
    return !( interval1 < interval2 );
}
```

Կասենք, որ առաջին ինտերվալը բացարձակ փոքր է երկրորդից, եթե նրա աջ ծայրակետի ինդեքսը փոքր է երկրորդ ինտերվալի ձախ ծայրակետի ինդեքսից: Պարզ ասած, երբ ինտերվալները չեն հատվում և առաջին ինտերվալը գտնվում է երկրորդից ձախ:

```
bool operator<<( const IntervalData& interval1, const IntervalData& interval2 )
{
    size_t nEnd1 = *( interval1.m_setCols.rbegin() ),
        nStart2 = *( interval2.m_setCols.begin() );

    return nEnd1 < nStart2;
}
```

Կասենք, որ առաջին ինտերվալը բացարձակ մեծ է երկրորդից, եթե նրա ձախ ծայրակետի ինդեքսը մեծ է երկրորդ ինտերվալի աջ ծայրակետի ինդեքսից: Պարզ ասած, երբ ինտերվալները չեն հատվում և առաջին ինտերվալը գտնվում է երկրորդից աջ: Նկատենք, որ «>>» գործողությունը «<<» գործողության մաքուր հակադարձը չէ:

```
bool operator>>( const IntervalData& interval1, const IntervalData& interval2 )
{
    size_t nStart1 = *( interval1.m_setCols.begin() ),
        nEnd2 = *( interval2.m_setCols.rbegin() );

    return nStart1 > nEnd2;
}
```

§ 3.2 Ինտերվալների գրաֆի արտապատկերումը C++ դասի միջոցով

Այժմ երբ սահմանել ենք IntervalData դասի օբյեկտների միջև բոլոր համեմատության գործողությունները կարող ենք նկարագրել IntervalGraph դասը:

IntervalGraph դասը կառուցվում է վերին և ստորին ուղիկների ոչ բացասական նիշերը պարունակող վեկտորների միջոցով, համապատասխանաբար՝ m_vecTop և m_vecBottom: Այս դասը մշակում է վեկտորների զույգը, ստանում համապատասխան ինտերվալները և դասավորում ինտերվալները համապատասխան տողերում:

```
class IntervalGraph
{
    using VecOfValues = std::vector< size_t >;
    using VecOfIntervals = std::vector< IntervalData >;
    using MultimapRowInterval = std::multimap< size_t, IntervalData >;
    ...
public:
    IntervalGraph( const VecOfValues& vecTop, const VecOfValues& vecBottom );
    ...
protected:
    void preprocessTopAndBottomVectors();
    void InitializeIntervals();

    VecOfMultisetIntervals separateIntervalsByRank();
    void determineRowsOfIntervals( VecOfMultisetIntervals& vecOfSetIntervals );

    const MultimapRowInterval& getMapIntervals() const { return m_mapRowInterval; }
    ...
protected:
    VecOfValues m_vecTop;
    VecOfValues m_vecBottom;
    VecOfIntervals m_vecIntervals;
    MultimapRowInterval m_mapRowInterval;
};
```

Կառուցիչը սկզբում preprocessTopAndBottomVectors ֆունկցիայի միջոցով համապատասխանեցնում է վերին և ստորին վեկտորները խնդրի պահանջներին, հետո InitializeIntervals ֆունկցիայի միջոցով դրանցից ստանում է ինտերվալների բազմություն:

Ստացված բազմությունը բաժանում է ավելի փոքր ենթաբազմությունների ըստ ուղղահայաց սահմանափակումների VG գրաֆում՝ յուրաքանչյուր ինտերվալին համապատասխանող գագաթին տանող ամենաերկար ճանապարհի երկարության (`IntervalData::m_nRank`):
 Ենթաբազմությունները ստանալուց հետո ստացված ենթաբազմությունների ինտերվալները դասավորում է համապատասխան տողերում:

```
IntervalGraph::IntervalGraph( const VecOfValues& vecTop,
                               const VecOfValues& vecBottom )
    : m_vecTop( vecTop )
    , m_vecBottom( vecBottom )
{
    // Ինտերվալ գրաֆը կառուցվում է ըստ վերին և ստորին ոտիկների նիշերի

    // Այն բոլոր նիշերը, որոնք պարունակվում են միայն վերին կամ միայն
    // ստորին վեկտորներում՝ փոխարինենք 0-ներով, քանի որ նրանք չպետք է մշակվեն
    preprocessTopAndBottomVectors();

    // Վերին և ստորին ոտիկների նիշերի արժեքների հիման վրա կառուցել
    // ինտերվալները և ավելացնել նրանց վեկտորում
    InitializeIntervals();

    // Ինտերվալների բազմությունը բաժանել առանձին ենթաբազմությունների ըստ իրենց
    // m_nRank-ի ( $VG$  գրաֆում դեպի գագաթը տանող ամենաերկար ճանապարհի)
    VecOfMultisetIntervals vecOfSetOfIntervals = separateIntervalsByRank();

    // Որոշել ինտերվալների տեղադրման տողերը
    // Նույն m_nRank-ով, չհատվող ինտերվալները կարող են տեղադրվել նույն տողում
    determineRowsOfIntervals( vecOfSetOfIntervals );
}
```

§ 3.3 Ինտերվալների գրաֆի կառուցումը

Այժմ մանրամասն ուսումնասիրենք IntervalGraph-ի օբյեկտի կառուցման վերոնշյալ չորս մասերը:

Ինդրի պահանջներին բավարարելու համար պետք է անտեսենք այն բոլոր նիշերը, որոնք պարունակվում են կա՛մ միայն վերին, կա՛մ միայն ստորին վեկտորներում: Անտեսելու համար preprocessTopAndBottomVectors ֆունկցիայում այդ նիշերը փոխարինում ենք 0-ական արժեքներով:

```
void IntervalGraph::preprocessTopAndBottomVectors()
{
    SetOfValues setTop( m_vecTop.begin(), m_vecTop.end() ),
                    setBottom( m_vecBottom.begin(), m_vecBottom.end() );

    for( size_t nVertexVal : setTop )
    {
        // Հեռացնել ${nVertexVal} արժեքով տարրը from the ${setBottom},
        // եթե այն պարունակվում է ն՛ վերին, ն՛ ստորին վեկտորներում
        if( setBottom.find( nVertexVal ) != setBottom.end() )
            setBottom.erase( nVertexVal );
        // Եթե ${nVertexVal} չի գտնվել ստորին վեկտորում, ապա
        // փոխարինել այն 0-ով՝ հետագայում անտեսելու նպատակով
        else
            std::replace( m_vecTop.begin(), m_vecTop.end(), nVertexVal, size_t(0) );
    }

    // ${setBottom}-ից արդեն հեռացվել են այն բոլոր տարրերը, որոնք պարունակվում են
    // վերին վեկտորում, հետևաբար մնացած տարրերը, եթե այդպիսիք կան, ավելորդ են:
    // Պետք է բոլորը փոխարինել 0-ով:
    for( size_t nVertexVal : setBottom )
    {
        std::replace( m_vecBottom.begin(), m_vecBottom.end(), nVertexVal, size_t(0) );
    }
}
```


Դիտարկենք այն ֆունկցիաների հայտարարությունները, որոնք օգտագործվում են InitializeIntervals ֆունկցիայում:

```
class IntervalGraph
{
public:
    ...
    void InitializeIntervals();

    void addTopVertexes( IntervalData& intervalCur, size_t nCurIndex );
    bool addBottomVertexes( IntervalData& intervalCur );
    ...
protected:
    VecOfValues m_vecTop;
    VecOfValues m_vecBottom;
    VecOfIntervals m_vecIntervals;
    MultimapRowInterval m_mapRowInterval;
};
```

Քանի որ ըստ խնդրի պահանջի ցանկացած ինտերվալ u վերին, u ստորին տողերում պետք է ունենա գոնե մեկ ոտիկ, ակնհայտ է, որ բոլոր ինտերվալները ստեղծելու համար անհրաժեշտ է և բավարար անցնել վերևի տողում գտնվող բոլոր ոտիկների վրայով և ստեղծել այդ ոտիկների նիշերին համապատասխան ինտերվալները:

Նկատենք, որ վերևի տողում կարող են հանդիպել նույն նիշով մեկից ավելի ոտիկներ: Այդ նիշի առաջին պատահույթի ժամանակ նրա բոլոր պատահույթները մշակվում են, այդ պատճառով հաջորդ քայլերում արդեն մշակված ինտերվալների նիշերով գազաթները կանտեսենք: Ծրագրի իրականացման համար վերոնշյալ արդեն մշակված ինտերվալների նիշերը կպահենք “setProcessedCols” անունով ոչ բացասական տարրերից բաղկացած կարգավորված բազմության (std::set<size_t>) մեջ:

Այսպիսով for ցիկլի միջոցով կանցնենք բոլոր վերին գազաթների վրայով: Ամեն քայլում կստուգենք, թե արդյոք այդ գազաթը չի գտնվում “setProcessedCols” բազմության մեջ: Եթե այն գտնվի, ապա կնշանակի, որ այդ նիշով ինտերվալն արդեն մշակվել է և կանցնենք հաջորդ քայլին: Հակառակ դեպքում, երբ այդ ոտիկի նիշը հանդիպում է առաջին անգամ

կավելացնենք այն “setProcessedCols” բազմության մեջ՝ հաջորդ քայլերում այդ նիշով ոտիկներն անտեսելու համար: Հաջորդիվ կստեղծենք ինտերվալը, ու վերին և ստորին տողերում կփնտրենք այդ նիշով ոտիկներ՝ ավելացնելով գտնված ոտիկների ինդեքսները ստեղծվող ինտերվալի ոտիկների (IntervalData::m_setCols) բազմության մեջ: Վերջում բոլոր գործողությունները հաջողությամբ ավարտելու դեպքում՝ ինտերվալը կպահպանենք «IntervalGraph::m_vecIntervals» բազմության մեջ:

```
void IntervalGraph::InitializeIntervals()
{
    SetOfValues setProcessedCols = { 0 };

    for( size_t i = 0; i < m_vecTop.size(); ++i )
    {
        int nVertexVal = m_vecTop[i];
        // Եթե ${nVertexVal} նիշով ինտերվալն արդեն մշակվել է, անցնել հաջորդ քայլին
        if( setProcessedCols.find( nVertexVal ) != setProcessedCols.end() )
            continue;

        setProcessedCols.emplace( nVertexVal );

        size_t nIndex = m_vecIntervals.size();
        IntervalData intervalCur( nIndex, nVertexVal );

        // Ինտերվալի նիշերի բազմությանն ավելացնել վերին տողում գտնվող նիշերը
        addTopVertexes( intervalCur, i );
        // Ինտերվալի նիշերի բազմությանն ավելացնել վերին տողում գտնվող նիշերը
        bool bAdd = addBottomVertexes( intervalCur );

        // Պահպանել ինտերվալը միայն այն դեպքում, երբ համապատասխան նիշերը
        // գտնվել են և՛ վերին, և՛ ստորին տողերում
        if( bAdd )
            m_vecIntervals.emplace_back( intervalCur );
    }
}
```

Քանի որ արդեն համոզվել ենք, որ վերևի տողում տվյալ նիշով ոտիկի առաջին պատահույթն ենք մշակում, վերևում այդ նիշով ոտիկ փնտրելիս՝ ավելորդ քայլերից խուսափելու համար ոտիկները կփնտրենք ընթացիկ ոտիկից դեպի աջ գտնվողների մեջ:

Բացի ինտերվալին պատկանող ոտիկների ինդեքսները հավաքելուց, նաև կհավաքենք տեղեկություն VG ուղղահայաց սահմանափակումների գրաֆում տվյալ ինտերվալին համապատասխան գագաթից դուրս եկող կողմնորոշված կողերի մասին:

Հիշեցում. Ուղղահայաց սահմանափակումների $VG = (V, E')$ գրաֆը կողմնորոշված գրաֆ է, որի յուրաքանչյուր $I \in VG$ գագաթը համապատասխանում է i համարով շղթային և յուրաքանչյուր j սյուն, որի համար $t(j)$ -ն և $b(j)$ -ն ոչ զրոյական թվեր են, առաջացնում է $t(j)$ -ից $b(j)$ կողմնորոշված կող:

```
void IntervalGraph::addTopVertexes( IntervalData& intervalCur, size_t nCurIndex )
{
    // Վերին տողում փնտրել ընթացիկ նիշով ոտիկները
    // Սկսել  $\{nCurIndex\}$ -երորդ ոտիկից, քանի որ նախորդներն արդեն մշակվել են
    for( size_t i = nCurIndex; i < m_vecTop.size(); ++i )
    {
        if ( m_vecTop[i] == intervalCur.getVertexValue() )
        {
            // Ավելացնել ոտիկի ինդեքսը ինտերվալի m_setCols զանգվածին
            intervalCur.addCol( i );

            // Ավելացնել  $VG$  գրաֆում ինտերվալին համապատասխանող գագաթից
            // դուրս եկող ուղղորդված կողին կից գագաթի մասին տեղեկություն
            if ( m_vecBottom[i] != 0 && m_vecBottom[i] != m_vecTop[i] )
            {
                intervalCur.addOutConnection( m_vecBottom[i] );
            }
        }
    }
}
```

Ներքևի տողում տվյալ նիշով գագաթ փնտրելիս կանցնենք բոլոր ոտիկների վրայով և նույն նիշով ոտիկ գտնելու դեպքում կավելացնենք ինտերվալի ոտիկների բազմությանը: Եթե ներքևի տողում տվյալ նիշով գագաթ չգտնվի, ապա կխախտվի պայմանը, որի համաձայն ցանկացած ինտերվալ պետք է ունենա նվազագույնը մեկ ստորին ոտիկ: Այդ դեպքում կվերադարձնենք տրամաբանական ոչ արժեքը, որն էլ «InitializeIntervals» ֆունկցիայում կխափանի ինտերվալի պահպանումը:

```

bool IntervalGraph::addBottomVertexes( IntervalData& intervalCur )
{
    bool bThereIsAtLeastOneBottom = false;

    // Ստորին տողում փնտրել ընթացիկ նիշով ուղիները
    for( size_t i = 0; i < m_vecBottom.size(); ++i )
    {
        if( m_vecBottom[i] == intervalCur.getVertexValue() )
        {
            // Ստորին գագաթում գտնվել է գոնե մեկ գագաթ: Ինտերվալը պետք է պահպանել
            bThereIsAtLeastOneBottom = true;
            // Ավելացնել ուղիկի ինդեքսը ինտերվալի m_setCols զանգվածին
            intervalCur.addCol( i );
        }
    }
    return bThereIsAtLeastOneBottom;
}

```

Ինտերվալների բազմությունը ստանալուց հետո `separateIntervalsByRank` ֆունկցիայի միջոցով այն կբաժանենք ենթաբազմությունների ըստ VG -ում ամենաերկար ճանապարհի երկարության: Այս գործողությունը կատարելու համար կօգտվենք VG -ի հարևանության մատրիցից, որտեղ 1 արժեք կունենան այն բոլոր տարրերը որի սյանը համապատասխանող գագաթից դուրս է գալիս կողմնորոշված կող դեպի նրա տողին համապատասխանող գագաթ ($\overrightarrow{v_j v_i}$, $\text{elem}[i][j] := 1$):

Դեպի i ինդեքսով գագաթ տանող ամենաերկար ճանապարհի երկարությունը կորոշենք A_{ij} հարևանության մատրիցը աստիճան բարձրացնելով: Նրա արժեքը կլինի այն k -րդ քայլի համարը, որում A_{ij}^k -ում i -րդ գագաթին համապատասխանող A_i^k տողը կդառնա 0-ական:

```

IntervalGraph::VecOfMultisetIntervals IntervalGraph::separateIntervalsByRank()
{
    MatrixOfValues matrixAdjacency = getAdjacencyMatrix();

    // Ստանալ բոլոր ինդեքսների բազմությունը { 1, 2, 3, ..., n }
    SetOfValues setNonZeroIndexes = getSetOfIntervalIndexes();
    // Ստեղծել 0-ական վեկտոր մատրիցի տողերի հետ համեմատելու համար
    VecOfValues vecOfZeros = VecOfValues( m_vecIntervals.size(), 0 );

    // Առաջին քայլով կվերցնենք միավոր մատրից, որպեսզի ցիկլի ամեն քայլում ունենանք
    // համապատասխան աստիճան բարձրացված հարևանության մատրից
    MatrixOfValues matrixPowAdjacency = getIdentityMatrix();

    VecOfMultisetIntervals vecOfSetOfIntervals;
    int nRank = -1;

    // Հարևանության մատրիցը աստիճան բարձրացնենք, քանի դեռ այն չի դառնա 0-ական մատրից
    while( !Algorithms::isMatrixOfZeros( matrixPowAdjacency ) )
    {
        ++nRank;
        vecOfSetOfIntervals.emplace_back();
        matrixPowAdjacency = Algorithms::multiplyMatrixes( matrixPowAdjacency, matrixAdjacency );

        // Ստուգել, այն բոլոր տողերը, որոնք 0-ական չէին նախորդ քայլում
        for( auto it = setNonZeroIndexes.begin(); it != setNonZeroIndexes.end(); )
        {
            size_t nCurIndex = *it;
            // Եթե տողը դարձել է 0-ական վեկտոր, ապա  $VG$ - ում դեպի նրան համապատասխան
            // զագաթ տանող ճանապարհի երկարությունը  $\{nRank\}$  է
            if( matrixPowAdjacency[ nCurIndex ] == vecOfZeros )
            {
                m_vecIntervals[ nCurIndex ].setRank( nRank );
                vecOfSetOfIntervals[ nRank ].emplace( m_vecIntervals[ nCurIndex ] );

                // հեռացնել ընթացիկ էլեմենտը և it-ին վերագրել նրա հաջորդին
                it = setNonZeroIndexes.erase( it );
            }
            else
                ++it;
        }
    }
    return vecOfSetOfIntervals;
}

```

Ինտերվալները համապատասխան տողերում տեղադրելու համար իրագործում ենք §2.4-ում նկարագրված VG -ի ներկման ալգորիթը:

```
void IntervalGraph::determineRowsOfIntervals(VecOfMultisetIntervals& vecOfSetIntervals )
{
    int nRow = 0;

    while( !vecOfSetIntervals.empty() )
    {
        auto& setIntervals = vecOfSetIntervals[0];

        // Անցնել յուրաքանչյուր ինտերվալի վրայով և որոշել նրա տողը կապուղում
        while ( !setIntervals.empty() )
        {
            IntervalData processedInterval = *( setIntervals.begin() );
            // Այս պահին ընթացիկ տողը դատարկ է, այդ պատճառով առանց ստուգումների
            // կարելի է ինտերվալը տեղադրել  $\{nRow\}$ -երորդ տողում
            m_mapRowInterval.emplace( nRow, processedInterval );
            setIntervals.erase( processedInterval );

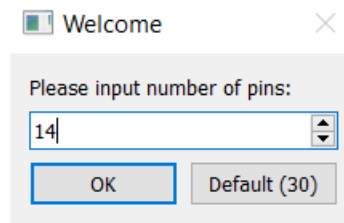
            IterMultisetIntervals it = setIntervals.begin();

            // Տեղադրել բոլոր հնարավոր ինտերվալները  $\{nRow\}$ -երորդ տողում:
            // Նույն տողում գտնվող ինտերվալները չպետք է հատվեն, հետևաբար ընթացիկ
            // տողում ամեն հաջորդ ինտերվալ, պետք է բացարձակ մեծ լինի նախորդից:
            // Հնարավորինս շատ ինտերվալներ տեղավորելու համար, պետք է ամեն քայլում
            // նախորդ ինտերվալից բացարձակ մեծերի բազմությունից ընտրել փոքրագույն
            // ինտերվալը: Քանի որ ինտերվալների բազմությունը կարգավորված է ըստ աճման,
            // ամեն քայլում ինտերվալը կփնտրենք ոչ թե սկսած առաջին ինտերվալից այլ
            // նախորդ քայլում գտնվածից:
            while( ( it = findFirstAbsoluteBiggerThan( it, setIntervals.end(), processedInterval ) )
                != setIntervals.end() )
            {
                m_mapRowInterval.emplace( nRow, *it );
                processedInterval = *it;
                it = setIntervals.erase( it );
            }
            ++nRow;
        }
        vecOfSetIntervals.erase( vecOfSetIntervals.begin() );
    }
}
```

Այսպիսով ստացանք `m_mapRowInterval` կառուցվածքը, որը նկարագրում է յուրաքանչյուր ինտերվալի տողը: Այս կառուցվածքը կփոխանցենք գրաֆիկական միջերեսն (GUI) ապահովող ծրագրային մասին, որն էլ կնկարի ստացված արդյունքը:

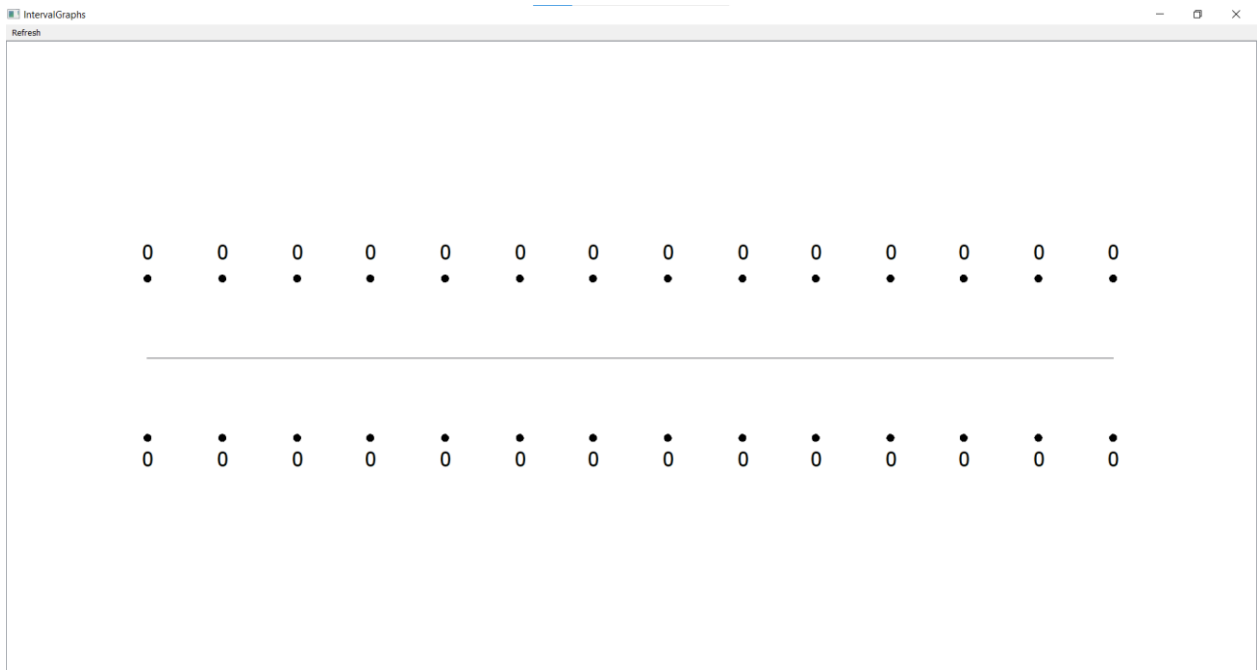
§ 3.4. Օգտագործման ուղեցույց

Ծրագիրը բացելիս հետևյալ «Welcome» պատուհանով օգտատիրոջը հնարավորություն է տրվում ընտրել վերին և ստորին ոտիկների քանակը:

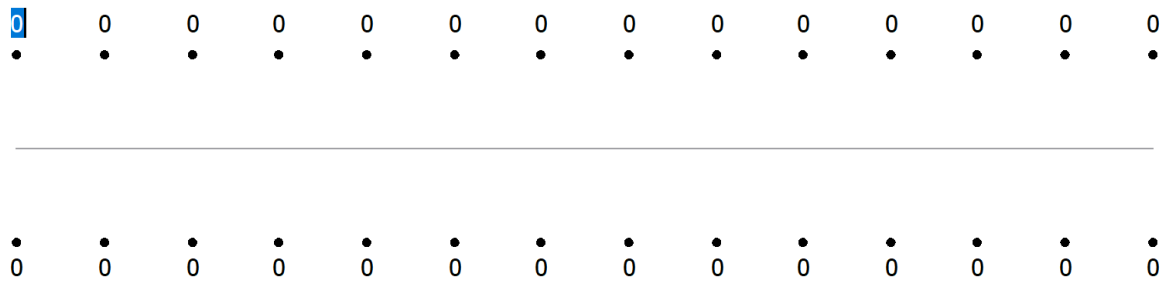


Այս պատուհանում $[5;1000]$ միջակայքի ցանկացած թիվ ընտրելուց հետո պետք է սեղմել «OK» կոճակը: Հնարավոր է նաև ընտրել լռելյայն 30 արժեքը՝ սեղմելով «Default (30)» կոճակը:

Ոտիկների քանակը որոշելուց հետո ծրագիրը կառուցում է որոշված քանակով վերին և ստորին 0 արժեքով ոտիկների բազմություն և կառավարումը հանձնում օգտատիրոջը:



Ոտիկներին համապատասխան արժեքները կարելի է փոխել՝ արժեքի վրա մկնիկի ձախ սեղմակի կրկնակի սեղմումով (double click):

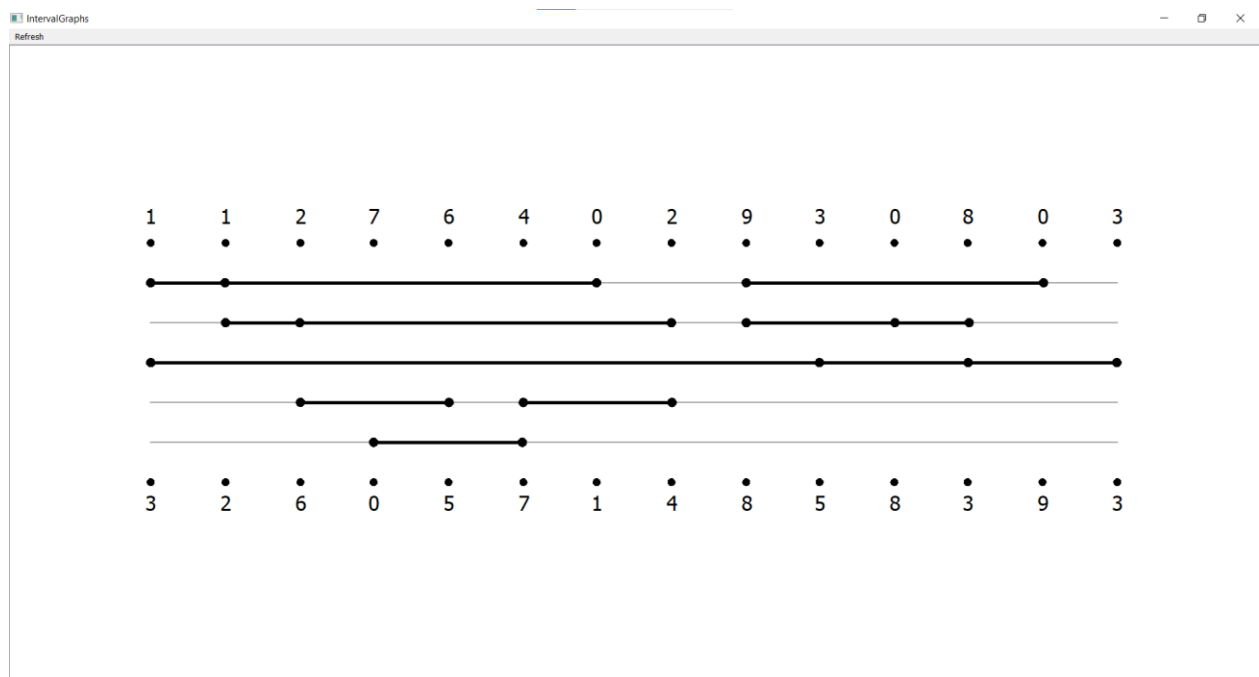


Հաջորդ տարրը հնարավոր է նաև ընտրել առանց մկնիկի օգնության՝ սեղմելով «Tab» կոճակը:

Ոտիկներին իրենց համապատասխան արժեքները վերագրելուց հետո, պետք է սեղմել «Refresh» կոճակը, որպեսզի կառուցվի համապատասխան ուղեգծումը:



Այն սեղմելուց հետո կստանանք հետևյալ պատկերը.



Արդյունքը ստանալուց հետո, ոտիկների արժեքները կարելի է փոփոխել և սեղմելով «Refresh» կոճակը ստանալ նոր արդյունքը:

Հարկ է նշել, որ ծրագիրը նախատեսված չէ VG գրաֆում կողմնորոշված ցիկլ պարունակող օրինակների համար: Այդպիսի օրինակների ներմուծման դեպքում ծրագրի պահվածքը հնարավոր է չհամապատասխանի սպասելիքներին:

ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ

Այսպիսով ավարտական աշխատանքում թարգմանվել է մասնագիտական գրականությունը, ներկայացվել է կապուղու ուղեգծման որոշակի խնդիր: Մշակվել է ալգորիթմ այդ խնդրի լուծման համար, որը ծրագրավորվել է C++ ծրագրավորման լեզվով: Մշակվել է համապատասխան գրաֆիկական միջերես (GUI) ծրագրի օգտագործման համար: Հաջողությամբ փորձարկվել է կոնկրետ օրինակների վրա: Փորձարկման արդյունքներն ու ծրագրային կոդը կարող եք գտնել «Հավելված» բաժնում:

ՕԳՏԱԳՈՐԾՎԱԾ ԳՐԱԿԱՆՈՒԹՅՈՒՆ

1. Велютин В.А., "Автоматизированное проектирование топологии БИС", Радио и связь, 112 с. 1983
2. Селютин В.А., "Машинное конструирование электронных устройств", Сов. радио, 384 с. 1977
3. Абрайтис Л.Б., Шейнаускас Р.И., Жилевичус В.А., "Автоматизация проектирования ЭВМ", Сов. радио, 272 с. 1979
4. Петросян А. В., Маркосян С. Е., Шукурян Ю. Г. "Математические вопросы автоматизации проектирования ЭВМ", Ереван, 144 с. 1977
5. Karapetyan I. A., "On the Restrictive Channel Thickness Estimation", Facta Univ. Ser.: Elec. Energ., vol. 20, No. 3, December 2007, pp. 499-506

ԳԼՈՒԽ 4. ԱՐՏԱԴՐԱՆՔԻ ԻՆՔՆԱՐԺԵՔԻ ԵՎ ՄԻԱՎՈՐ ԳՆԻ ՀԱՇՎԱՐԿ

Արտադրանքի (աշխատանքների, ծառայությունների մատուցման) ինքնարժեքը արտադրանքի թողարկման (աշխատանքների իրականացման և ծառայությունների մատուցման) գործընթացում ծախսվող հումքի, նյութերի, վառելիքի, հիմնական միջոցների ամորտազիացիայի, աշխատանքային ռեսուրսների վարձատրման, ինչպես նաև դրանց արտադրության և իրացման հետ կապված այլ ծախսերի արժեքային գնահատականն է:

Արտադրանքի ինքնարժեքի մեջ մտնող ծախսերը դասակարգվում են ըստ տնտեսական տարրերի և ըստ կալկուլյացիոն հոդվածների:

Ներկայումս կիրառվում է ծախսերի ըստ կալկուլյացիոն հիմնական հոդվածների հետևյալ դասակարգումը՝

1. հումք և նյութեր,
2. գնովի կիսաֆաբրիկատներ և համալրող առարկաներ,
3. էներգիա և վառելանյութ տեխնոլոգիական նպատակների համար,
4. արտադրության մեջ անմիջականորեն զբաղված բանվորների հիմնական աշխատավարձ,
5. արտադրության մեջ անմիջականորեն զբաղված լրացուցիչ աշխատավարձ,
6. սարքավորումների շահագործման և պահպանման ծախսեր,
7. ընդհանուր արտադրական ծախսեր: արտադրամասային ինքնարժեք,
8. ընդհանուր տնտեսական ծախսեր,
9. արտադրական այլ ծախսեր, արտադրական ծախսեր,
10. արտաարտադրական ծախսեր, լրիվ ինքնարժեք:

Սարքը, որի ինքնարժեքն ու միավորի գինը ենթակա է որոշման, ռադիոընդունիչ է:

Հաշվարկի համար ելքային տվյալներն են հանդիսանում՝

- սարքավորման թողարկման համար ծախսվող հումքը և նյութերի անվանացանկը, միավոր արտադրանքին ընկնող ծախսերի նորմատիվը, կիլոգրամի/գրամի գինը,
- արտադրվող սարքավորման քանակը՝ ընդունել 10000 միավոր:
- սարքավորման մեջ մտնող գնովի կիսաֆաբրիկատների ու համալրող առարկաների քանակն ու անվանացանկը,
- ժամանակի ամփոփ նորմերը, աշխատանքի կարգն ու աշխատավարձի ձևերը,
- ժամավճարային և գործարքային պարգևատրման չափերը (24%),
- լրցուցիչ աշխատավարձի չափերը (14%),
- սարքավորումների պահպանման և շահագործման ծախսերի դրույքաչափերը,
- ընդհանուր արտադրական ծախսերի դրույքաչափը (98%),
- ընդհանուր տնտեսական ծախսերի դրույքաչափը (120%)
- ոչ արտադրական ծախսերի դրույքաչափը (արտադրանքի ինքնարժեքի 2%-ը):

§ 4.1. Հումքի և նյութերի ծախսի որոշումը

Հիմնական հումքի և նյութերի արժեքը որոշվում է հետևյալ բանաձևով՝

$$\mathcal{O}_{\text{հն}} = \sum \mathcal{P}_i \cdot \mathcal{Q}_i,$$

որտեղ \mathcal{P}_i -ն i -րդ տեսակի նյութի ծավալն է (կգ/գ), \mathcal{Q}_i - 1 կգ/գ i -րդ տեսակի նյութի գինը (դրամ):
Հաշվարկային արդյունքները բերված են աղյուսակ 1-ում:

Աղյուսակ 1

Նյութի տեսակն ու անվանումը	Նյութի ծախս, կգ	Միավոր նյութի գինը, դրամ	Սարքի նյութի արժեքը, դրամ
Զոդ ПOC-78	0,15	1 750	262
Պողպատ CT-5	0,27	1 300	351
Ֆլուս ФКС	0,025	400	10
Ապակե մանրաթել Фкс1-30	0,1	550	55
Լաք YP-317	0,1	3 000	300
Սպիրտ	0,08	300	240
Ընդամենը	-	-	1 218

Ընդամենը՝ $1218 \cdot 10\,000 = 12\,180\,000$ դրամ:

§ 4.2. Գնովի կիսաֆաբրիկատների և համալրող առարկաների ծախսի հաշվարկ

Գնված բաղադրիչների և կիսաֆաբրիկատների արժեքը որոշվում է հետևյալ բանաձևով՝

$$\Sigma_{կֆհա} = \Sigma \text{Բ}_i * \text{Գ}_i,$$

որտեղ Բ_i -ն i -րդ տեսակի գնված բաղադրիչների քանակն է, հատ, Գ_i -ն i -րդ գնված բաղադրիչի միավորի գինը, դրամ: Հաշվարկի արդյունքները բերված են աղյուսակ 2-ում:

Աղյուսակ 2

Բաղադրիչի անվանումն ու տեսակը	Մեկ սարքին ընկնող քանակ, հատ	Միավորի գինը, դրամ	Մեկ սարքին ընկնող արժեք, դրամ
Միկրոսխեմա	1	2 200	2 200
Տրանզիստոր	10	700	7 000
Դիոդ	18	50	900
Ռեզիստոր MJT0,25	8	100	800
Ալեհավաք	1	2 250	2 250
Ենթակառուցվածքային ռեզիստոր	4	200	800
Էլեկտրոնային կոնդեսատորներ	26	80	2 080
Միակցիչներ	6	50	300
Տրանսֆորմատորներ	1	800	800
Ապահովիչներ	1	250	250
Ապահովիչի պահպանիչ	1	500	500
Կոճակներ	5	150	750
Ընդհամենը	-	-	15 165

Ընդամենը՝ $15\,165 * 10\,000 = 151\,650\,000$ դրամ:

§ 4.3. Էլեկտրաէներգիայի ծախսի հաշվարկ

Ուժային սարքավորումները և հաստոցները աշխատեցնելու համար էլեկտաէներգիայի տարեկան ծախսը որոշվում է հետևյալ բանաձևով՝

$$E = \varphi \cdot X \cdot U_{\text{է}},$$

որտեղ φ -ն էլեկտրաէներգիայի տարեկան ծախսն է, $U_{\text{է}}$ -ն 1 կվ/ժ էլեկտրաէներգիայի արժեքն է (50 դրամ, ըստ գործարանային գների)

$$\varphi = Z_{\text{տհ}} \cdot \Phi_{\text{հ}} \cdot \varphi_{\text{բ}} \cdot \varphi_{\text{ւ}} / \varphi_1 \cdot \varphi_2,$$

որտեղ $Z_{\text{տհ}}$ -ն հաստոցների էլեկտրաշարժիչների ընդհանուր հզորությունն է և $Z = 1$ կՎտ,

$\Phi_{\text{հ}}$ -ն՝ հաստոցների աշխատաժամանակի տարեկան իրական ֆոնդը՝ 25 օր,

$\varphi_{\text{բ}}$ -ն՝ հաստոցների բեռնվածության միջին գործակիցը՝ 0.9,

$\varphi_{\text{ւ}}$ -ն՝ կորուստներն են ցանցում՝ 0.8,

φ_1 -ը՝ միաժամանակ աշխատող հաստոցների շարժիչի Օ.Գ.Գ.՝ 0.95,

φ_2 -ը՝ շարժիչի Օ.Գ.Գ.-ն՝ 0.9,

$$\varphi_{\text{բ}} = 0.9, \varphi_{\text{ւ}} = 0.8, \varphi_1 = 0.95, \varphi_2 = 0.9,$$

$$\varphi = 1000 / 0.95 \cdot 0.9 = 1169,5 \text{ կՎտ/ժ},$$

$$E = 1169,5 \cdot 50 \approx 50\,000 \text{ դրամ/ամսական և } 50\,000 \cdot 12 = 600\,000 \text{ դրամ/տարեկան:}$$

§ 4.4. Արտադրական բանվորների հիմնական աշխատավարձի հաշվարկը

Արտադրական բանվորների հիմնական աշխատավարձի մեջ մտնում են՝

- գործարքային դրույքաչափերով աշխատավարձը,
- ժամավճարային աշխատավարձը,
- պարգևավճարը:

Գործարքային աշխատավարձն ըստ տարիֆային համակարգի որոշվում է հետևյալ բանաձևով՝

$$U_{\text{հիմ.}} = \sigma_{\text{դ.}} \cdot U_{\text{արտ.}},$$

որտեղ $\sigma_{\text{դ.}}$ -ն ժամային դրույքաչափն է, $U_{\text{արտ.}}$ -ն՝ արտադրանքի աշխատատարությունը, ժամային նորմը:

Հաշվարկման արդյունքները բերված են աղյուսակ 3-ում:

Գործառույթի հաջորդական- ություն	Վճարման ձև	Աշխատանքի կարգը	Ժամանա- կային նորմ	Ժամային դրույք	Տարիֆային Ֆոնդ
Նախապատ- րաստում	Գործարքա- պարգևա- վճարային	4	26	3 500	91 000
Մշակում	Գործարքա- պարգևա- վճարային	3	30	5 000	150 000
Կարգավորում	Գործարքա- պարգևա- վճարային	4	25	5 000	125 000
Տեղակայում	Գործարքա- պարգևա- վճարային	4	30	4 500	135 000
Ընդամենը	-	-	-	-	501 000

Պարգևատրման չափը որոշվում է հետևյալ բանաձևով՝

$$\mathcal{N} = \mathcal{U}_{\text{հիմ}} * \mathcal{N}_{\eta} / 100\%,$$

որտեղ \mathcal{N}_{η} -ն պարգևատրման դրույքաչափն է (%):

$$\mathcal{N} = 501\,000 * 24 / 100\% = 120\,240 \text{ դրամ: (1)}$$

Ընդամենը հիմնական աշխատավարձը կկազմի՝

$$501\,000 + 120\,240 = 621\,240 \text{ դրամ/ամսական կամ՝ (2)}$$

$$621\,240 * 12 = 7\,454\,880 \text{ դրամ/տարեկան: (3)}$$

§ 4.5. Արտադրական բանվորների լրացուցիչ աշխատավարձի հաշվարկը

Լրացուցիչ աշխատավարձի մեջ մտնում են՝ հերթական և լրացուցիչ գործողումների, արձակուրդների վճարները, պետական հանձնարարականների կատարման հետ կապված ծախսերը և այլն: Արտադրական աշխատակիցների լրացուցիչ աշխատավարձը հաշվարկվում է հետևյալ բանաձևով՝

$$U_{\text{լր.}} = \text{Ը}U_{\text{հիմ.}} * U_{\text{լր.դ.}}/100, (4)$$

որտեղ

$\text{Ը}U_{\text{լր.դ.}}$ –ն ընդհանուր հիմնական աշխատավարձն է, իսկ

$U_{\text{լր.դ.}}$ –ն՝ լրացուցիչ աշխատավարձի դրույքաչափը (%):

Մեր օրինակի համար՝

$$U_{\text{լր.}} = (501\,000 + 120\,240) * 14/100 = 86\,959.6 \text{ դրամ/ամսական կամ՝} \\ 86\,959.6 * 12 = 1\,043\,515.2 \text{ դրամ/տարեկան: (5)}$$

4.6. Սարքավորումների պահպանման և շահագործման ծախսերի հաշվարկը

Սարքավորումների պահպանման և շահագործման ծախսերի թվին են պատկանում ամորտիզացիոն, ընթացիկ վերանորոգման, տրանզիտորային միջոցների, գործիքների և հարմարանքների վերանորոգման և այլ ծախսերը: Սարքավորումների պահպանման և շահագործման ծախսերի թվին են պատկանում ամորտիզացիոն, ընթացիկ վերանորոգման, տրանզիտորային միջոցների, գործիքների և հարմարանքների վերանորոգման և այլ ծախսերը:

Հիմնական արտադրական միջոցների տարեկան ամորտիզացիան (U_s) հաշվարկվում է հետևյալ բանաձևով՝

$$U_s = Z_u / \text{Ն},$$

որտեղ Z_u -ն հիմնական միջոցների սկզբնական արժեքն է, Ն-ն՝ հիմնական միջոցների օգտակար գործունեության ժամկետը:

Հիմնական միջոցի անվանումն ու տեսակը	Հիմնական միջոցի սկզբնական արժեքը	Ամորտիզացիոն հատկացումներ	
		ՀՄ օգտակար գործողության ժամկետ, տարի	Ամորտիզացիոն ծախս
Արտադրական շենքեր, շինություններ	321 000 000	20	16 050 000
Հաստոցներ, սարքավորումներ	46 250 000	5	9 250 000
Ընդամենը	-	-	25 300 000

Սարքավորումների պահպանման և շահագործման ծախսերի գումարը որոշվում է հետևյալ կերպ՝

$$\bar{\sigma}_{\text{պ.շ.}} = U_{\text{հիմ}} * \bar{\sigma}_{\text{պ.շ.դ.}} / 100,$$

որտեղ

$\bar{\sigma}_{\text{պ.շ.դ.}}$ -ն սարքավորումների պահպանման և շահագործման ծախսերի դրույքաչափն է, $U_{\text{հիմ}}$ -ը՝ արտադրական բանվորների տարեկան հիմնական աշխատավարձը: Մեր օրինակի համար՝

$$\bar{\sigma}_{\text{պ.շ.}} = 7\,454\,880 * 0,016 = 119\,278.08 \text{ դրամ: (6)}$$

Սարքավորումների ընթացիկ վերանորոգման ծախսը որոշվում է հետևյալ բանաձևով՝

$$\bar{\sigma}_{\text{ը.վ.}} = U_{\text{հիմ}} * \bar{\sigma}_{\text{ը.վ.դ.}} / 100,$$

որտեղ $\bar{\sigma}_{\text{ը.վ.դ.}}$ -ն սարքավորումների ընթացիկ վերանորոգման դրույքաչափն է: Այն կկազմի՝

$$\bar{\sigma}_{\text{ը.վ.}} = 7\,454\,880 * 0,07 = 521\,841.6 \text{ դրամ: (7)}$$

Աղյուսակ 5

Ծախսի անվանումը	Դրույքաչափը	Տարեկան ծախսը
Սարքավորումների պահպանման և շահագործման ծախսեր	1.6%	119 278.08
Սարքավորումների ընթացիկ վերանորոգում	7%	521 841.6
Սարքավորումների ամորտիզացիա	-	25 300 000
Ընդամենը	-	25 941 119.68

§ 4.7. Ընդհանուր արտադրական ծախսերի հաշվարկը

Ընդհանուր արտադրական ծախսերի մեջ մտնում են առանձին արտադրական կառույցների, ամորտիզացիոն, պահպանման և ընթացիկ վերանորոգման, խնայողությունների և հայտնագործությունների, վերահսկողական և այլ աշխատանքների հետ կապված ծախսերը: Այս ծախսերը որոշվում են հետևյալ բանաձևով՝

$$\text{ԸԱԾ} = U_{\text{հիմ}} * \text{ԸԱԾ} / 100,$$

որտեղ ԸԱԾ-ն ընդհանուր արտադրական ծախսերի տոկոսն է: Մեր օրինակի համար՝

$$\text{ԸԱԾ} = 7\,454\,880 * 98 / 100 = 7\,305\,782.4 \text{ դրամ:} \quad (8)$$

§ 4.8. Ընդհանուր տնտեսական ծախսերի հաշվարկը

Ընդհանուր տնտեսավարման ծախսերի մեջ մտնում են ձեռնարկության ընդհանուր կառավարման-վարչական՝ գործարանը կառավարող անձնակազմի աշխատավարձի, գործուղման, տպագրական, փոստային-հեռագրային ծախսերը և այլ ծախսեր: Այն որոշվում է հետևյալ բանաձևով՝

$$\text{ԸՏԾ} = U_{\text{հիմ}} * \text{ԸՏԾ} / 100,$$

որտեղ ԸՏԾ -ն ընդհանուր տնտեսավարման ծախսերի տոկոսն է:

$$\text{ԸՏԾ} = 7\,454\,880 * 120 / 100 = 8\,945\,856 \text{ դրամ:} \quad (9)$$

§ 4.9. Արտաարտադրական ծախսերի հաշվարկը

Արտաարտադրական ծախսերի մեջ մտնում են փաթեթավորման, արտադրանքի տեղափոխման, գովազդի և այլ ծախսեր: Այն հաշվարկվում են հետևյալ կերպ՝

$$\text{ԱԱԾ} = \text{Ի}_{\text{արտ.}} * \text{ԱԱԾ}_{\text{տկ}},$$

որտեղ $\text{Ի}_{\text{արտ.}}$ -ն արտադրական ինքնարժեքն է, $\text{ԱԱԾ}_{\text{տկ}}$ -ն՝ արտադրականի տեսակարար կշիռը ինքնարժեքի մեջ: Մեր օրինակի համար՝

$$\text{ԱԱԾ} = 215\,121\,152 * 0,025 = 5\,378\,028 \text{ դրամ:} \quad (10)$$

§ 4.10. Սարքի ընդհանուր ինքնարժեքի կալկուլյացիան

Լրիվ ինքնարժեքի կալկուլյացիան բերված է աղյուսակ 5-ում:

Աղյուսակ 5

N	Ծախսերի հոդվածի անվանումը	գումարը, դրամ
1.	Հումք և նյութեր	12 180 000
2.	Գնովի կիսաֆաբրիկատներ և համալրող առարկաներ	151 650 000
3.	Էներգիա և վառելանյութ	600 000
4.	Հիմնական աշխատավարձ	7 454 880
5.	Լրացուցիչ աշխատավարձ	1 043 515.2
6.	Սարքավորումների պահպանման և շահագործման ծախսեր	25 941 119.68
7.	Ընդհանուր արտադրական ծախսեր	7 305 782
	Արտադրամասային ինքնարժեք	206 175 296
8.	Ընդհանուր տնտեսական ծախսեր	8 945 856
	Արտադրական ինքնարժեք	215 121 152
9	Արտաարտադրական ծախսեր	5 378 029
	Լրիվ ինքնարժեք	220 499 181

Սարքավորման միավորի լրիվ ինքնարժեքը կկազմի՝

$$\Gamma_{\text{լրիվ}} = 220\,499\,181 / 10000 = 22\,049.9181 \text{ դրամ: (11)}$$

§ 4.11. Շահույթի և միավորի գնի հաշվարկը

Սարքի գնի հաշվարկը իրականացվում է արտադրական ծախսերին ուղղված մեթոդով: Ձեռնարկության գինն իր մեջ ընդգրկում է ձեռնարկության շահույթն ու լրիվ ինքնարժեքը: Ձեռնարկության շահույթի հաշվարկն իրականացվում է հետևյալ բանաձևով՝

$$\mathcal{C} = \mathcal{H}_{\text{լրիվ}} * \mathcal{C} / 100,$$

որտեղ $\mathcal{H}_{\text{լրիվ}}$ –ը սարքավորման լրիվ ինքնարժեքն է, \mathcal{C} –ն ձեռնարկության շահույթի դրույքաչափը՝ տոկոսներով արտահայտված: Մեր օրինակի համար՝

$$\mathcal{C} = 22\,049.9181 * 13 / 100 = 2\,866.4893 \text{ դրամ: } (12)$$

Ձեռնարկության գնի հաշվարկը իրականացվում է հետևյալ կերպ՝

$$\mathcal{Q}_{\Delta\text{են}} = \mathcal{H}_{\text{լրիվ}} + \mathcal{C}:$$

Մեր օրինակի համար՝

$$\mathcal{Q}_{\Delta\text{են}} = 22\,049.9181 + 2\,866.4893 = 24\,916.4074 \approx 24\,916 \text{ դրամ: } (13)$$

Սարքավորման բացթողնման գինը որոշվում է հետևյալ բանաձևով՝

$$\mathcal{Q}_{\text{բաց.}} = \mathcal{Q}_{\Delta\text{են}} + \text{ԱԱՀ},$$

որտեղ $\mathcal{Q}_{\text{բաց.}}$ –ը սարքավորման բացթողնման գինն է, ԱԱՀ–ն՝ ավելացված արժեքի հարկը (20%).

$$\mathcal{Q}_{\text{բաց.}} = 24\,916 + 24\,916 * 20 / 100 = 29\,899 \text{ դրամ: } (14)$$

Այսպիսով, կատարված հաշվարկների արդյունքում ստացանք, որ ռադիոընդունիչ սարքավորման միավորի լրիվ ինքնարժեքը կկազմի 22 050 դրամ, իսկ գինը՝ 29 899 դրամ:

**ԳԼՈՒԽ 5. K 200-130 ԷՆԵՐԳԱԲԼՈՎՆԵՐՈՎ ԿԱՀԱՎՈՐՎԱԾ ՋԷԿ-Ի
ԾԽԱԳԱԶԵՐԻ ՀԵՌԱՑՄԱՆ ԾԽՆԵԼՈՒՅԶԻ ԲԱՐՁՐՈՒԹՅԱՆ
ՀԱՇՎԱՐԿ**

Էներգետիկայի զարգացման հետևանքով մեծանում է շրջակա միջավայր արտանետվող վնասակար նյութերի ներգործությունը: Կաթսայական տեղակայանքների ծխնելույզները նախատեսված են ծխագազերի հեռացման համար: Ծխնելույզները կառուցվում են մետաղից կամ աղյուսից: Էներգետիկայի զարգացմանը զուգընթաց ծխնելույզներն սկսեցին օգտագործել ծխագազերում պարունակվող վնասակար խառնուրդների՝ մինչև թույլատրելի խտությունները մթնոլորտում ցրելու նպատակով: ՋԷԿ-երի հզորության մեծացմանը զուգընթաց առաջացավ զգալի բարձր (200, 300մ և ավելի) ծխնելույզների կառուցման անհրաժեշտությունը:

Այս առումով ծխնելույզների շինարարության ոլորտում առաջադիմական լուծում էր միաձույլ երկաթբետոնյա փողերի կիրառումը, որոնք դիմակայում էին բավական զգալի կշռային և քանու բեռնվածքներին: Միաժամանակ անհրաժեշտ էր ընդունել, որ զգալի ամրությամբ օժտված երկաթբետոնե կոնստրուկցիան բավական թույլ գտնվեց ծխագազերում պարունակվող ծծմբային միացությունների, խոնավության և բարձր ջերմաստիճանային պայմանների նկատմամբ: Զգացվեց ծխնելույզների երկրորդ, ներքին թաղանթի ստեղծման անհրաժեշտություն, ինչը պետք է չեզոքացներ հիմնական փողի վրա ծխագազերի ագրեսիվ ազդեցությունը:

Ծխնելույզի երկաթբետոնե փողի ներքին մակերևույթը պատվում է մեկուսիչով, որը ներկայացնում է էպոքսիդային լաք և ապակեթելային գործվածք: Վերջինիս հաջորդում է երկաթբետոնե հեծանների վրա (յուրաքանչյուր 30...40մ մեկ) տեղադրվող կարմիր և թթվակայուն աղյուսաինդեզիտային մածուկով ամրացված շերտը, որը կոչվում է աղյուսապատվածք:

Ծխնելույզի երկաթբետոնե փողի պաշտպանության նպատակով նրա վերին կտրվածքի վրա տեղադրվում է թուջե գլխադիր:

Ծխնելույզի վրա, հատուկ լուսահարթակներում, տեղադրված են ըստ ծխնելույզի բարձրության 15 կամ 30մ-ը մեկ, լուսազդանշանային համակարգեր: Ծխնելույզն ըստ բարձրության 15 միջակայքով ներկում են 2 կամ 2.5մ լայնությամբ կարմիր գույնի շերտով:

Ծխնելույզը կահավորված է նաև շանթարգելային համակարգով: Էլեկտրակայանի շահագործման ամբողջ ժամանակահատվածում (30...50 տարի) ծխնելույզը պետք է աշխատի առանց վերանորոգման: Ոչ կիպ աղյուսապատվածքի դեպքում նրա միջով ներթափանցում են ագրեսիվ գազեր, ինչպես նաև ծխագազերում պարունակվող գոլորշու արդյունքում տեղի է ունենում ծխնելույզի փողի աստիճանական քայքայման գործընթաց: Այս առումով բավականաչափ լավ արդյունք է տալիս ծխնելույզի փողի կրող և գազահեռացուցիչ գործառույթների տարանջատումը: Վերջինիս իրականացման համար փողի կրող կոնստրուկցիայի և գազահեռացուցիչ կանալի (անցքի) միջև ստեղծվում է 150-200 մմ տրամագծով օդային բացակ:

Այդ բացակում ապահովվում է օդի շրջանառություն՝ ոչ կիպ աղյուսապատվածքից ներթափանցած գազերի հեռացման համար: Օդը նախապես տաքացվում է մինչև 50...90°C տուրբինից վերցված շոգու ջերմության հաշվին:

Տարածված է նաև այսպես կոչված <<խողովակը խողովակի մեջ>> տիպի ծխնելույզները, որոնց փողերի միջև եղած զգալի բացակը մարդկանց տեղաշարժի և վերանորոգման աշխատանքների հնարավորություն է տալիս: Այդ դեպքում գազահեռացուցիչ փողը բաղկացած է օդային տարրերից, որոնցից յուրաքանչյուրի բարձրությունը 10մ է, կազմված մետաղական կախովի հիմնականախքներից, որոնց վրա ամրացվում են սիլիցիումաբետոնային սալեր: Վերջիններս օժտված են հակակոռոզիոն հատկությամբ: Տարատեսակ մեծաթիվ սարքավորումներով կահավորված ՋԷԿ-երում, հատկապես ջերմա-էլեկտրակենտրոններում, կիրառություն են ստացել նաև բազմափող ծխնելույզները: Այս դեպքում երկաթբետոնե փողի ներսում տեղադրվում է (սովորաբար 3-4) մետաղական զլանաձև փողեր:

Բազմափող ծխնելույզների հիմնական առավելությունը յուրաքանչյուր փողի՝ մյուսներից անկախ սպասարկման և վերանորոգման հնարավորությունն է:

§ 8.1. Ծխնելույզի նվազագույն թույլատրելի բարձրության հաշվարկը

Էներգետիկական տեղակայանքների նախագծման և սարքավորումների ընտրման կարևորագույն բաղադրիչներից է այրման արգասիքների հուսալի հեռացման և նրանցում պարունակվող վնասակար նյութերի արդյունավետ ցրման սարքերի՝ ծխնելույզների բարձրության հաշվարկը:

Ծխնելույզի բարձրությունը որոշվում է ըստ հետևյալ արտահայտության՝

$$h_{\text{ծխ}} = \sqrt{\frac{m \cdot A \cdot M \cdot F \cdot n \cdot \eta}{U \cdot \rho \cdot \gamma - C_{\Phi}}} \cdot \sqrt[3]{\frac{z}{V_1 \cdot \Delta T}}, \quad (6.1)$$

որտեղ՝

A-ն մթնոլորտի ջերմաստիճանային շերտաբաշխման (ուղղաձիգ հարթությունում օդի ջերմաստիճանի բաշխումը) գործակիցն է անբարենպաստ օդերևութաբանական պայմաններում (Հայաստանի համար A=200),

M-ն աղտոտող նյութերի գումարային արտանետումն է, գ/վ,

V₁-ը ծխնելույզից արտանետվող ծխագազերի ծավալային ծախսն է, մ³/վ,

ΔT-ն արտանետվող ծխագազերի և շրջակա միջավայրի օդի ջերմաստիճանների տարբերությունն է, °C,

m-ը և n-ը չափում չունեցող գործակիցներ են, որոնք կախված են ծխնելույզի ելքում ծխագազերի արագությունից (ՋԷԿ-ի համար n=1, m-ը վերցնում ենք ըստ աղյուսակ 6.1-ի),

F-ը չափում չունեցող մեծություն է, որը հաշվի է առնում մթնոլորտում վնասակար նյութերի նստեցման արագությունը (գազային արտանետումների համար F=1),

z-ը միևնույն բարձրության ծխնելույզների թիվը (z=1),

η-ն՝ չափում չունեցող մեծություն է, որը հաշվի է առնում տվյալ տեղանքի ռելիեֆի ազդեցությունը (հարթ և ոչ շատ կտրտված տեղանքի համար η=1),

C_Φ-ը գործող կազմակերպությունների և տրանսպորտի աշխատանքով պայմանավորված աղտոտող նյութերի ֆոնային խտությունն է տվյալ շրջանում, (C_Φ=0 մգ/մ³),

ՄԹԽ-ն աղտոտող նյութերի սահմանային թույլատրելի խտությունն է, ծծմբի երկօքսիդի համար $ՄԹԽ=0,5 \text{ մգ/մ}^3$, ազոտի օքսիդի համար $ՄԹԽ=0,085 \text{ մգ/մ}^3$:

Նախկին ԽՍՀՄ-ի առանձին շրջանների համար հաստատված են A գործակցի հետևյալ մեծությունները՝

- Միջին Ասիայի մերձարևադարձային գոտիներ և Չինաստանի մարզ՝ 250,
- Ղազախստան, Ստորին Պովոլժիե, Կովկաս, Մոլդովա, Սիբիր, Հեռավոր Արևելք և Միջին Ասիա՝ 200,
- նախկին ԽՍՀՄ-ի եվրոպական մասի հյուսիս և հյուսիս-արևմուտք, Միջին Պովոլժիե, Ուրալ և Ուկրաինա՝ 160,
- նախկին ԽՍՀՄ-ի եվրոպական մասի կենտրոն՝ 140:

Ջերմաստիճանային ΔT տարբերությունը հաշվարկելիս անհրաժեշտ է նկատի ունենալ, որ շրջակա միջավայրի օդի ջերմաստիճանը վերցվում է ըստ կլիմայական տեղեկատուի՝ ըստ վայրի ամենաշոգ ամսվա միջին կեսօրյա ջերմաստիճանի:

$$\Delta T = T_{\text{դիւ}} - T_{\text{միջ}} = 140 - 30 = 110^{\circ}\text{C}:$$

m գործակցի արժեքները կախված ծխնելույզի ելքում ծխազագերի արագությունից (W_0)

Աղյուսակ 6.1

W_0 , մ/վ	10 ... 15	20 ... 25	30 ... 35
M	1	0.9	0.8

M-ը SO_2 -ի և NO_2 -ի գումարային արտանետումներն են և դրանք կարելի է որոշել հետևյալ արտահայտությունների միջոցով՝

$$M_{SO_2} = 2 \cdot S^P \cdot (1 - \eta'_{SO_2}) \cdot B_{24}, \quad (6.2)$$

որտեղ՝

S^P -ն վառելիքի բանվորական զանգվածում ծծմբի պարունակությունն է, $S^P = 0.85\%$,

B_{24} -ն վառելիքի ծախսն է շոգեկաթսայում, $B_{24} = 12.03 \text{ կգ/վ}$:

η'_{SO_2} -ը կաթսայի գազանցքերում թռչող մոխրի որսման չափն է (հեղուկ վառելիքի դեպքում այն կարելի է վերցնել հավասար 0.02, իսկ կարծր վառելիքի համար այն տատանվում է 0.1...0.5-ի սահմաններում):

$$M_{SO_2} = 2 \cdot \frac{0.85}{100} \cdot (1 - 0.02) \cdot 12.03 \cdot 3 \cdot 1000 = 601.3 \text{ գ/վ},$$

$$M_{NO_2} = 0.34 \cdot 10^{-7} \cdot \beta \cdot K \cdot B_{24} \cdot Q_u^p \cdot \left(1 - \frac{q_4}{100}\right) (1 - \varepsilon \cdot r), \quad (6.3)$$

որտեղ՝

K -ն ազոտի օքսիդների ելքը բնութագրող գործակիցն է, վերագրված պայմանական վառելիքին, կգ/տ, (կգ/մ³),

B_{24} -ն բնական վառելիքի լրիվ ծախսն է, գ/վ (տ/ժ),

q_4 -ը վառելիքի մեխանիկական թերայրման հետևանքով ջերմության կորուստներն են, ($q_4 = 0.5$),

β -ն ուղղման գործակից է, որը հաշվի է առնում վառելիքում ազոտի պարունակության ազդեցությունը ազոտի օքսիդների ելքի վրա ($\beta = 0.9$),

ε -ը գործակից է, որը հաշվի է առնում վերաշրջանառվող գազերի ազդեցության արդյունավետությունը, կախված նրանց հնոց մատուցվելու պայմաններից ($\varepsilon = 0.02$),

r -ը ծխագազերի վերաշրջանառության աստիճանն է, % ($r = 15$),

Q_u^p -ն վառելիքի այրման ջերմությունն է, $Q_u^p = 40560$ կՋ/կգ,

70տ/ժ շոգեարտադրողականությունից բարձր շոգեարտադրողականություն ունեցող կաթսաներում հեղուկ և գազային վառելիքների այրման ժամանակ, բեռնվածքների ողջ միջակայքում՝

$$K = \frac{12 \cdot D}{200 + D}, \quad (6.4)$$

$$K = \frac{12 \cdot 640}{200 + 640} = 9.14 \text{ կգ/տ},$$

որտեղ D -ն կաթսայի անվանական շոգեարտադրողականությունն է, $D = 640$ տ/ժ:

$$M_{NO_2} = 0.34 \cdot 10^{-7} \cdot 9.14 \cdot 12.03 \cdot 3 \cdot 10^3 \cdot 40560 \cdot \left(1 - \frac{0.5}{100}\right) \cdot 0.9 \cdot (1 - 0.02 \cdot 15) = 285.15 \text{ գ/վ}:$$

Որոշում ենք ծծմբի և ազոտի օքսիդների գումարային արտանետումների քանակը՝

$$M = M_{SO_2} + \frac{U_{\text{թ}^{\text{IV}}_{SO_2}}}{U_{\text{թ}^{\text{IV}}_{NO_2}}} \cdot M_{NO_2} \quad (6.5)$$

ընդ որում՝

$$U\theta\text{Խ}_{\text{SO}_2} = 0.5\text{մգ/մ}^3, U\theta\text{Խ}_{\text{NO}_2} = 0.085\text{մգ/մ}^3 :$$

$$M = 601,3 + \frac{0.5}{0.085} \cdot 285.15 = 2278.65\text{գ/վ} :$$

V_1 -ը, որոշվում է հետևյալ արտահայտությամբ.

$$V_1 = n \cdot V_{\delta\text{խ}} \quad (6.6)$$

որտեղ՝

$V_{\delta\text{խ}}$ -ն յուրաքանչյուր էներգաբլոկի այրման արգասիքների ծավալային ծախսն է, ($V_{\delta\text{խ}} = 275.5\text{մ}^3/\text{վ}$),

n -ը՝ էներգաբլոկների թիվը, $n = 3$:

$$V_1 = 3 \cdot 275,5 = 826.5\text{մ}^3/\text{վ} :$$

Ստացված տվյալների հիման վրա կարելի է հաշվել ծխնեկույզի բարձրությունը՝

$$h_{\delta\text{խ}} = \sqrt{\frac{200 \cdot 0.9 \cdot 2278.65 \cdot 1 \cdot 1 \cdot 1}{0.5}} \sqrt{\frac{1}{826.5 \cdot 110}} = 135.1\text{մ}$$

Ելնելով ծխնեկույզի հաշվարկված արժեքից, ըստ շինարարական ստանդարտի, ընտրում ենք $h=150\text{մ}$ բարձրությամբ և $D_F=7.2\text{մ}$ բկի տրամագծով ծխնեկույզ:

Ստուգում:

Ընտրված ծխնեկույզի պարագայում որոշում ենք ելքում ծխագազերի հնարավոր ծավալային ծախսը:

$$V_{\delta\text{խ}} = \frac{\pi \cdot d_p^2}{4} \cdot W_0 \text{ մ}^3/\text{վ}, \quad (6.7)$$

$$V_{\delta\text{խ}} = \frac{3.14 \cdot (7.2)^2}{4} \cdot 25 = 1017.36 \text{ մ}^3/\text{վ}:$$

Քանի որ $V_{\delta\text{խ}} > V_1$, հետևաբար ընտրված ծխնեկույզը կբավարարի ծխագազերի հեռացմանը:

§ 8.2. Ծրագրի իրականացումը C++ ծրագրավորման լեզվով

```
#include <iostream>
#include <cmath>

int main()
{
    int A = 200, n = 1, nCount = 3,
        F = 1, z = 1, ghetta = 1,
        deltaT = 110, C_f = 0, Q_s__b = 40560;

    double STX_S0_2 = 0.5, STX_NO_2 = 0.085, V_cx = 275.5, m = 0.9, d_b = 7.2,
        T_b = 0.85 / 100, ghetta_SO_2 = 0.02, B_shk = 12.03, W_0 = 25,
        q_4 = 0.5, betta = 0.9, epsilon = 0.02, r = 15, K = 9.14;

    double M_SO_2 = 2 * T_b * ( 1 - ghetta_SO_2 ) * B_shk * nCount * 1000,
        M_NO_2 = 0.34 * std::pow( 10, -7 ) * betta * K * B_shk * nCount * 1000
            * Q_s__b * ( 1 - q_4 / 100 ) * ( 1 - epsilon * r ),
        M = M_SO_2 + ( STX_S0_2 / STX_NO_2 ) * M_NO_2,
        V_1 = nCount * V_cx,
        var1 = m * A * M * F * n * ghetta,
        var2 = STX_S0_2 - C_f,
        var3 = pow( ( z / V_1 * deltaT ), 1 / 3 ),
        h_cx = sqrt( var1 / var2 * var3 );

    std::cout << "h_cx: " << h_cx << std::endl;

    double V_cx_res = 3.14 * std::pow( d_b, 2 ) / 4 * W_0;

    if( V_cx_res > V_1 )
        std::cout << "Kbavarari" << std::endl;
    else
        std::cout << "Chi bavarari" << std::endl;
}
```


ԳԼՈՒԽ 6. ԼԱՁԵՐԱՅԻՆ ՃԱՌԱԳԱՅԹՄԱՆ ՕԳՏԱԳՈՐԾՈՒՄԻՑ ԱՌԱՋԱՑԱԾ ՎՏԱՆԳՆԵՐԻ ԼՈՒՍԱԲԱՆՈՒՄ

Լազերային ճառագայթումը էլեկտրամագնիսական ճառագայթման հատուկ տեսակ է, որն առաջանում է 0,2 ... 1000 մկմ ալիքի երկարության միջակայքում: Լազերային ճառագայթման հիմնական աղբյուրներն են քվանտային օպտիկական գեներատորները (COGs) և որոշ երևույթների (մետալուրգիա, ապակեգործություն) կողմնակի գործոնները: Լազերային ճառագայթման աղբյուրների հետ աշխատելիս անձնակազմը կարող է ենթարկվել ուլտրամանուշակագույն, տեսանելի և ինֆրակարմիր տիրույթներում բարձր ինտենսիվության ճառագայթման, ռենտգենյան ճառագայթների և ռադիոհաճախակա- նության ճառագայթման ազդեցության, բարձր էլեկտրական լարման ազդեցության, ինչպես նաև լազերային ճառագայթով սինթետիկ նյութեր մշակելիս օդի փոշոտության: Այնուամենայնիվ, հիմնական վնասակար գործոնը լազերային ճառագայթման ինտենսիվությունն է, որը լինում է 3 տեսակի՝ ուղղակի, արտացոլված և ցրված: Լազերային ճառագայթումը կարող է առաջանալ ալիքի երկարության 0.2-ից 1000 մկմ տիրույթում, որը բաժանված է հետևյալ սպեկտրալ շրջանների.

- Ուլտրամանուշակագույն - 0,2-ից 0,4 մկմ
- Տեսանելի - 0.4-ից 0.75 մկմ
- Ինֆրակարմիր մոտ - 0,75-ից 1,4 մկմ
- Հեռավոր ինֆրակարմիր - ավելի քան 1,4 մկմ

Լազերների կողմից առաջացած կարճ իմպուլսները (10^{-10} վ) կարող են վնասել աչքը շատ ավելի կարճ ժամանակահատվածում, քան անհրաժեշտ է պաշտպանիչ ֆիզիոլոգիական մեխանիզմների գործարկման համար (թարթման ռեֆլեքս 0,1 վ):

Ուլտրամանուշակագույն ճառագայթման վտանգը: Ուլտրամանուշակագույն ճառագայթումը արդյունաբերական պայմաններում առաջանում է էլեկտրական եռակցման, սնդիկ-քվարցային լամպերի գործողության, էլեկտրական վառարաններում մետաղի հալման ժամանակ, օգտագործվում է ֆիլմերի և լուսանկարչական արդյունաբերության մեջ, լուսապատճենահանման և պլազմային պրոցեսներում: Այն նաև օգտագործվում է ստորգետնյա հանքերում, ինչպես նաև ֆիզիոթերապիայի սենյակներում աշխատողների մոտ վիտամին D-ի պակասը լրացնելու համար:

Ուլտրամանուշակագույն ճառագայթման հիմնական արհեստական աղբյուրներն են բարձր և միջին ճնշման սնդիկի լամպերը, քսենոնային աղեղային լամպերը և տարբեր գազերի խառնուրդներ պարունակող լամպերը, որոնք ներառում են քսենոն կամ սնդիկի գոլորշի:

Ուլտրամանուշակագույն ճառագայթների կենսաբանական ակտիվությունը կախված է դրանց ալիքի երկարությունից: Կախված ալիքի երկարությունից ճառագայթման ազդեցությունը կլինի հետևյալը.

1. 0,2 - 0,28 միկրոն - ակտիվորեն գործում է հյուսվածքային սպիտակուցների և լիպիդների, և այլ մոլեկուլների վրա (ԴՆԹ, ՌՆԹ), որի արդյունքում կարող է առաջացնել հեմոլիզ:

2. 0,28 - 0,31 միկրոն - մաշկի վրա թողնում է ուժեղ ազդեցություն

3. 0.31 - 0.4 միկրոն - ուլտրամանուշակագույն A ճառագայթը թողնում է թույլ կենսաբանական ազդեցություն, որը հիմնականում վտանգավոր է եղջերաթաղանթի և ոսպնյակի համար: Մասնավորապես այս տիրույթում ուլտրամանուշակագույն լույսի չափազանց մեծ ազդեցությունը կարող է առաջացնել աչքի կարմրություն, բարձր արցունքարտադրություն, եղջերաթաղանթի մշուշվածություն և այլն: Այս ազդեցությունները սովորաբար առաջանում են ճառագայթումից հետո 24 ժամվա ընթացքում:

Լազերային ճառագայթման վտանգը տեսանելի և մոտ ինֆրակարմիր տիրույթում: Լազերային ճառագայթումը սպեկտրի տեսանելի շրջանում առաջացնում է տեղաշարժեր էնդոկրին և իմունային համակարգերի, կենտրոնական և ծայրամասային նյարդային համակարգերի, սպիտակուցների, ածխաջրերի և լիպիդային նյութափոխանակության մեջ: Այն 0,514 մկմ ալիքի երկարությամբ հանգեցնում է սիմպաթոադրենալ և հիպոֆիզի ադրենալ համակարգերի գործունեության փոփոխության: Մոտ ինֆրակարմիր տիրույթում(1,06 մկմ) ճառագայթման երկարատև քրոնիկական ազդեցությունը ալիքի երկարությամբ առաջացնում է վեգետատիվ-անոթային խանգարումներ: Գրեթե բոլոր հետազոտողները, ովքեր ուսումնասիրել են լազերներ սպասարկող անձանց առողջական վիճակը, շեշտում են, որ նրանց մոտ ավելի հաճախ են հայտնաբերվում ասթենիկ և վեգետատիվ-անոթային խանգարումներ:

Մոտ ինֆրակարմիր տիրույթում (0.4 մկմ – 1.4 մկմ) լազերային ճառագայթումները վտանգավոր են նաև աչքի ցանցաթաղանթի համար: Ինֆրակարմիր A ճառագայթը եղջերաթաղանթով փոխանցվում է աչքի ոսպնյակին, որն այն կենտրոնացնում է ցանցաթաղանթի վրա՝ ճառագայթման ազդեցությունը ուժեղացնելով մինչև 100 000 անգամ: Ցանցաթաղանթի վրա առաջացած վնասվածքները չեն բուժվում, քանի որ ցանցաթաղանթի հյուսվածքային կառուցվածքները անվերականգնելի են:

Լազերային ճառագայթման վտանգը հեռավոր ինֆրակարմիր տիրույթում: Հեռավոր ինֆրակարմիր տիրույթում (0,14 մկմ և ավելի) լազերային ճառագայթումը տաքացնում է արցունքները՝ այդպիսով վնասելով եղջերաթաղանթի հյուսվածքները: Ինֆրակարմիր ճառագայթման չափազանց մեծ ազդեցությունը հանգեցնում է եղջերաթաղանթի թափանցիկության կորստի կամ մակերևույթային անկանոնությունների:

Լազերային ճառագայթման ազդեցությունը մաշկի վրա: Լազերային ճառագայթումը վտանգ է ներկայացնում նաև մաշկի համար: Այն հիմնականում ի հայտ է գալիս այրվածքների տեսքով: Այդպիսի վնասը հիմնականում առաջանում է ուլտրամանուշակագույն կամ մոտ ինֆրակարմիր ալիքի երկարությամբ լազերի հետ 10 միկրովայրկյանից ավելի շփման դեպքում:

Լազերային ճառագայթման ջերմային ազդեցությունը հիմնականում կախված է հետևյալ գործոններից.

- Հյուսվածքների կլանման կարողություն
- Ճառագայթման ուժգնություն
- Ազդեցության տևողություն և պարբերականություն
- Ճառագայթված տարածքի մակերես

Միջին ինտենսիվության ճառագայթման դեպքում առաջանում են մաշկի ջերմային այրվածքներ: Լազերի և մաշկի շփման կետում ջերմաստիճանը կտրուկ բարձրանում է: Տեղի է ունենում ներքջջային հեղուկի եռում և գոլորշիացում: Մաշկը կարմրում է: Ճնշման տակ տեղի է ունենում հյուսվածքային կառուցվածքների խզում: Մաշկի վրա առաջանում է այտուց, որոշ դեպքերում՝ ներմաշկային արյունազեղումներ: Հետագայում այրվածքի տեղում հայտնվում են մահացած հյուսվածքներ:

Ինֆրակարմիր լազերային ճառագայթումը կարող է խորը ներթափանցել հյուսվածքների միջով և վնասել ներքին օրգանները: Խորը այրվածքի հատկանիշը առողջ և վնասված հյուսվածքների հերթափոխն է: Հատկանշականն այն է, որ ճառագայթման ենթարկվելիս մարդը ցավ չի զգում: Ինֆրակարմիր ճառագայթման ժամանակ ամենախոցելի օրգանը լյարդն է:

Ընդհանուր առմամբ մարմնի վրա ճառագայթման ազդեցությունը առաջացնում է կենտրոնական նյարդային համակարգի ֆունկցիոնալ խանգարումներ, սրտանոթային ակտիվություն՝ արյան ճնշման անկում, ուժեղացված քրտնարտադրություն, անբացատրելի ընդհանուր հոգնածություն, դյուրագրգռություն:

ՀԱՎԵԼՎԱԾ

Խնդրի իրագործման ծրագրային կոդն ու ծրագիրը կարող եք գտնել պահոցում՝ այցելելով հետևյալ հղումով կամ սկանավորելով հղմանը հաջորդող QR կոդը.

https://github.com/Erik-Asatryan/Restrictive_Channel_Routing.git

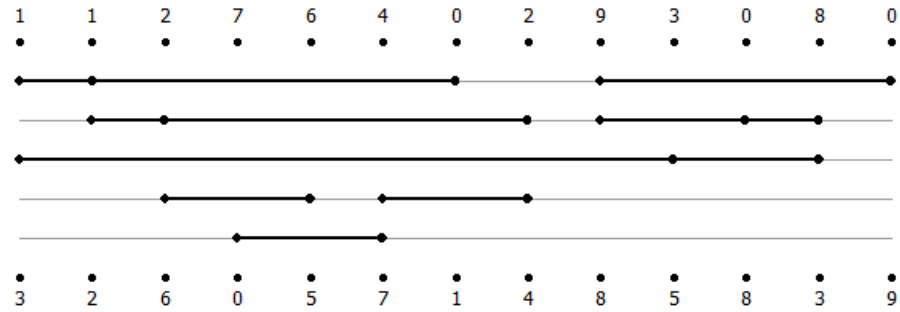


Ավելի մանրամասն տեղեկություն կարող եք գտնել ուղղորդված պահոցի «0_README» ֆայլում:

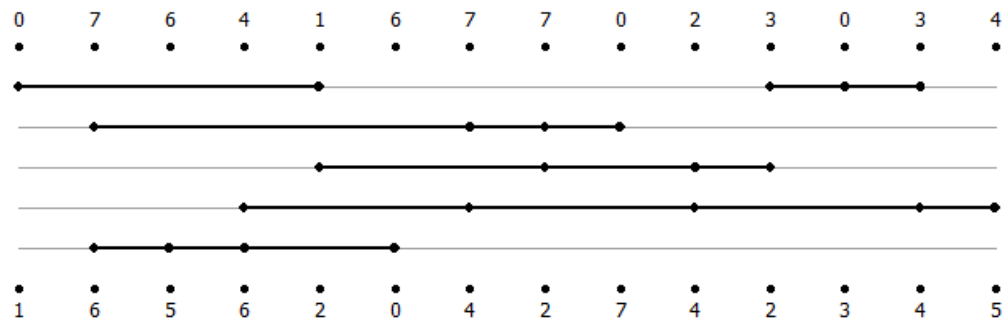
Թեստավորման արդյունքներ

Ներկայացնենք մի քանի փորձնական արդյունքներ.

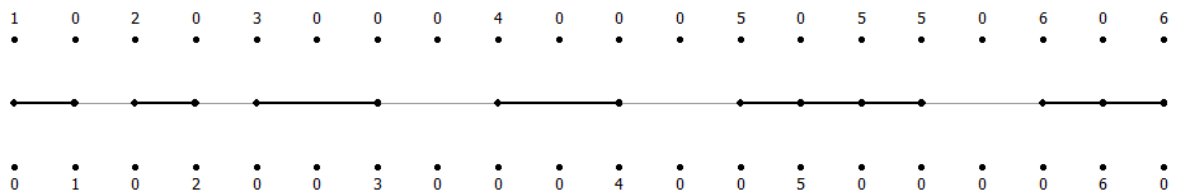
1.



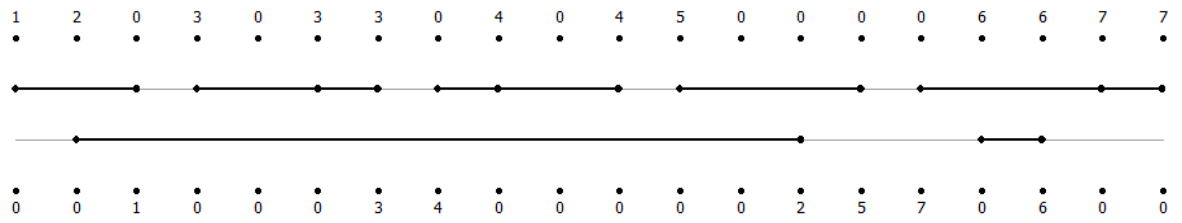
2.



3.



4.



5.

