Վ. Գաբրիելյան

ԿՈՄԲԻՆԱՏՈՐ ԱԼԳՈՐԻԹՄՆԵՐ ԵՎ ԱԼԳՈՐԻԹՄՆԵՐԻ ՎԵՐԼՈՒԾՈՒԹՅՈՒՆ

Դասախոսություններ

Գլուխ 1

ՈՐՈՆՄԱՆ ԼԱՎԱԳՈՒՅՆ ԱԼԳՈՐԻԹՄՆԵՐ

Դասընթացի հիմնական նպատակն է դիտարկել տարբեր բնույթի կոմբինատոր խնդիրներ, առաջարկել այդ խնդիրները լուծող ալգորիթմներ, գնահատել այդ ալգորիթմների բարդությունը, որից ելնելով՝ ընտրել նրանցից լավագույնը:

§ 1.1. ԲԱԶՄՈՒԹՅԱՆ ՏԱՐՐԻ ՈՐՈՆՈՒՄ

Որոշակի խնդիրները կամ որոշակի դասի խնդիրները լուծող ալգորիթմների համեմատության համար նախ անհարժեշտ է սահմանել ալգորիթմի բարդության գաղափարը (հասկացությունը): Այդ նպատակով քննարկենք մի օրինակ:

Դիցուք տրված են n գնդիկներ՝ համարակալված $1, 2, \ldots, n$ թվերով: Հայտնի է, որ այդ գնդիկներից մեկը ռադիոակտիվ է (թե որը՝ չգիտենք): Գոյություն ունի սարք, որը մեկ ստուգումով որոշում է ռադիոակտիվ գնդիկի առկայությունը մեր կողմից ընտրված գնդիկների ենթաբազմության մեջ: Պահանջվում է նվազագույն թվով նշված տիպի ստուգումների միջոցով որոշել ռադիոակտիվ գնդիկը:

Ձևակերպված խնդրի լուծման համար կարելի է առաջարկել տարբեր ալգորիթմներ: Դիտարկենք երկու այդպիսի ալգորիթմ:

1.1. *Ալգորիթմ:*

Մեկ առ մեկ, համարների $1, 2, \ldots, n$ հերթականությամբ, սարքի վրա ստուգում ենք գնդիկները ՝ մինչև ռադիոակտիվ գնդիկի գտնելը:

1.2. *Ալգորիթմ (կիսումների եղանակ):*

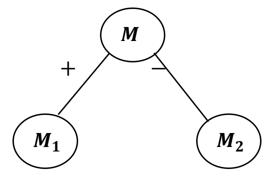
(i) *Որպես փորձարկման ենթակա գնդիկների բազմություն ընդունել $\{1, 2, \ldots, n\}$ բազմությունը:*

(ii) *Փորձարկման ենթակա գնդիկների բազմությունը տրոհել երկու համարյա հավասար մասերի (որոնցում գնդիկների քանակները տարբերվում են ամենաշատը մեկով) և սարքի վրա ստուգել ռադիոակտիվ գնդիկի առկայությունը նրանցից մեկում: Ռադիոակտիվ գնդիկը պարունակող ենթաբազմությունը ընդունել որպես փորձարկման ենթակա գնդիկների բազմություն:*

(iii) *Եթե փորձարկման ենթակա գնդիկների բազմությունը պարունակում է մեկ գնդիկ, ապա այդ գնդիկը ռադիոակտիվ է և ավարտել ալգորիթմի աշխատանքը, հակառակ դեպքում՝ վերադառնալ* (ii) *քայլի կատարմանը:*

Նկատենք, որ առաջարկված երկու ալգորիթմը լուծում են վերը նկարագրված խնդիրը, այսինքն՝ գտնում են ռադիոակտիվ գնդիկը: Նշենք նաև, որ ալգորիթմներից յուրաքանչյուրում ցանկացած ստուգում բնորոշվում է գնդիկների M ենթաբազմությամբ և ստուգման արդյունքով՝ ‹այո› կամ ‹ոչ› պատասխանով:

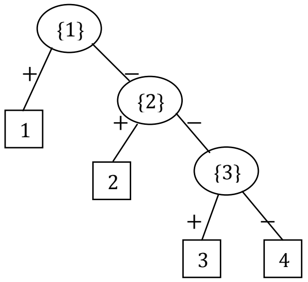
Ալգորիթմների համեմատության հետ կապված խնդիրների հստակեցման նպատակով, այդ ալգորիթմները կպատկերենք, այսպես կոչված, **որոնման ծառի** միջոցով: Ռադիոակտիվ գնդիկի որոնման խնդիրը լուծող ցանկացած ալգորիթմին համապատասխան որոնման ծառը հանդիսանում է **2-ծառ**, որի արմատին համապատասխանում է ալգորիթմում կատարվող առաջին ստուգումը: Հաջորդիվ, եթե որոնման ծառի որևէ գագաթի համապատասխանեցված է M ենթաբազմությունը բնորոշող ստուգումը, ապա այդ գագաթին հաջորդող գագաթներից մեկին համապատասխանեցնենք այն M_1 ստուգումը, որը կատարվում է ալգորիթմում, երբ M ստուգման արդյունքը ‹այո› է (նշելով նրան համապատասխան կողը), իսկ մյուս գագաթին համապատասխանեցնենք այն M_2 ստուգումը, որը կատարվում է ալգորիթմում, երբ M ստուգման արդյունքը ‹ոչ› է.



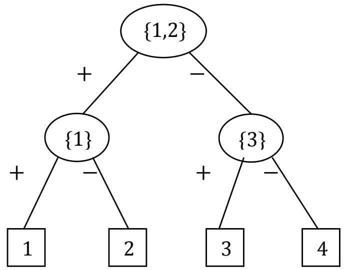
Նկար 1.1:

Ալգորիթմի աշխատանքի ավարտին համապատասխանող գագաթները կոչվում են որոնման ծառի **տերևներ:**

Հետևյալ երկու նկարներում համապատասխանաբար պատկերված են 1.1. ալգորիթմին և 1.2. ալգորիթմին համապատասխանող որոնման ծառերը $n = 4$ դեպքում:



Նկար 1.2:



Նկար 1.3:

Նկատենք, որ տերևի բարձրությունը կամ երկարությունը (այդ գագաթն արմատին միացնող ճանապարհի կողերի քանակը) ալգորիթմի կատարած ստուգումների քանակն է, երբ ռադիոակտիվ է այդ գագաթին համապատասխանող գնդիկը:

Նկարներից պարզ է, որ որոշ դեպքերում առաջին ալգորիթմն է (1.1. ալգորիթմ) քիչ ստուգումներով լուծում խնդիրը, որոշ դեպքերում էլ՝ երկրորդ ալգորիթմը (1.2. ալգորիթմ): Այս հանգամանքը դժվարեցնում է առաջարկված ալգորիթմների համեմատումը: Այդ դժվարության հաղթահարման ողջամիտ եղանակներից մեկը հետևյալն է. սահմանվում է ալգորիթմի կողմից պահանջվող ստուգումների միջին քանակ՝ հնարավոր դեպքերից յուրաքանչյուրի համար պահանջվող ստուգումների քանակների գումարը՝ բաժանած դեպքերի քանակի վրա: Յուրաքանչյուր ալգորիթմ բնութագրվում է պահանջվող ստուգումների միջին քանակով և գերադասելի է համարվում այն ալգորիթմը, որի համար այդ բնութագրիչը փոքր է:

Համաձայն ալգորիթմի բարդության այս սահմանմանը, $n = 4$ գնդիկների համար 1.1. ալգորիթմում պահանջվող ստուգումների միջին քանակը հավասար է 2.25, իսկ 1.2. ալգորիթմում՝ 2: Հետևաբար այդպիսի մոտեցման դեպքում 1.2. ալգորիթմը գերադասելի է 1.1. ալգորիթմից:

Չնայած նրան, որ ալգորիթմների բարդությունն ուսումնասիրող խնդիրներում գնահատման այս եղանակը հաճախ է կիրառվում, սակայն նրա հետ առընչվող մաթեմատիկական խնդիրները մեր դասընթացի շրջանակներից դուրս են և մենք կսահմանենք ալգորիթմի գնահատման այլ եղանակ:

Ռադիոակտիվ գնդիկի որոնման խնդրի լուծման ալգորիթմի բարդություն կհամարենք ռադիոակտիվ գնդիկը գտնելու համար՝ ալգորիթմում պահանջվող ստուգումների առավելագույն քանակը: Վերջինիս համաձայն, 1.1. ալգորիթմում $n = 4$ գնդիկների համար ստուգումների առավելագույն քանակը 3 է (երբ ռադիոակտիվ գնդիկն ունի 3 կամ 4 համարը), հետևաբար նրա բարդությունը հավասար է 3, իսկ 1.2. ալգորիթմի բարդությունը՝ 2:

1.3. ***Սահմանում:*** *Տրված խնդիրը լուծող **ալգորիթմի բարդություն** համարվում է խնդրի լուծման համար պահանջվող (անհրաժեշտ) թույլատրելի գործողությունների առավելագույն քանակը:*

Այսուհետև կդիտարկենք ալգորիթմի բարդության միայն այս տիպի գնահատական՝ վատագույն դեպքում գործողությունների քանակը, և պարզ է, որ այն՝ այդ ալգորիթմին համապատասխանող որոնման ծառի բարձրությունն է՝ ծառի ամենաերկար ճյուղի երկարությունը:

1.4. ***Սահմանում:*** *Տրված խնդիրը լուծող ալգորիթմը, որի բարդությունը՝ խնդիրը լուծող ուրիշ ալգորիթմների բարդությունների համեմատությամբ ամենափոքրն է, կոչվում է խնդիրը լուծող **լավագույն ալգորիթմ:***

Այժմ դիտարկվող օրինակի համար կարող ենք ձևակերպել օպտիմիզացիայի հետևյալ խնդիրը. **Գտնել ռադիոակտիվ գնդիկն որոնող լավագույն ալգորիթմ, այսինքն այնպիսի ալգորիթմ, որի բարդությունն ամենափոքրն է, կամ որ նույնն է, որում ստուգումների առավելագույն քանակն ամենափոքրն է:** Պարզ է, որ այդպիսի ալգորիթմ գոյություն ունի: Իսկապես, լավագույն ալգորիթմի գոյությունը հետևում է այն բանից, որ 1.1. ալգորիթմը ռադիոակտիվ գնդիկի որոնման խնդիրը լուծում է ոչ ավել քան $(n-1)$ ստուգմամբ, իսկ $(n-1)$-ից փոքր կամ հավասար բարձրություն ունեցող ծառերի քանակը վերջավոր է: Հետևաբար նրանց մեջ կգտնվի ամենափոքր բարձրություն ունեցող ծառը: Ռադիոակտիվ գնդիկն որոշող լավագույն ալգորիթմի բարդությունը նշանակենք $T_r(n)$:

Դիտարկենք բազմության տարրի որոնման խնդիրը: Դիցուք տրված է $A_n = \{a_1, a_2, \dots, a_n\}$ բազմությունը և $x \in A_n$ մեզ անհայտ տարրը: Թույլատրվում է x տարրի մասին տալ ‹այո› կամ ‹ոչ› պատասխան ունեցող ցանկացած հարց և ստանալ պատասխան: Խնդիրը հետևյալն է. **Գտնել x տարրն որոնող լավագույն ալգորիթմ, այսինքն այնպիսի ալգորիթմ, որում հարցերի առավելագույն քանակը լինի ամենափոքրը:** Հեշտ է նկատել, որ այս ձևակերպված խնդրի և նախորդ խնդրի տարբերությունն այն է, որ ռադիոակտիվ գնդիկն որոշելու դեպքում տրվում էին որոշակի $x \in M \subseteq \{1, 2, \dots, n\}$ տիպի հարցեր, իսկ այս դեպքում հարցերի բնույթի վրա սահմանափակում չի դրվում:

Բազմության տարրն որոնող լավագույն ալգորիթմի գոյությունն ակնհայտ է. խնդիրը լուծող ալգորիթմներին համապատասխան ծառերի քանակը վերջավոր է: Այդ ալգորիթմի բարդությունը նշանակենք $T_s(n)$:

1.5. *Լեմմա:* $T_s(n) \le T_r(n)$:

Ապացույց: Հեշտ է նկատել, որ ռադիոակտիվ տարրն որոնող լավագույն ալգորիթմում M ենթաբազմությանը համապատասխանող ստուգումը փոխարինենք $x \in \{a_i / i \in M\}$ հարցով, ապա կստանանք բազմության տարրն որոնող ալգորիթմ:

1.6. *Լեմմա:* $\lceil \log_2 n \rceil \le T_s(n)$:

Ապացույց: Բազմության տարրն որոնող լավագույն ալգորիթմի բարդությունը հավասար է $T_s(n)$, ուստի այդ ալգորիթմին համապատասխան որոնման 2-ծառի բարձրությունը նույնպես $T_s(n)$ է, և A_n բազմության ամեն մի $a_i, i = 1, 2, \dots, n$, տարրի կհամապատասխանի որոնման ծառի տերև՝ այն գագաթը, որում ալգորիթմն ավարտում է աշխատանքը, երբ $x = a_i$: A_n բազմության տարբեր տարրերի պետք է համապատասխանեն տարբեր տերևներ: Հետևաբար, որոնման ծառի տերևների քանակն առնվազն n է: Մյուս կողմից $T_s(n)$ բարձրություն ունեցող 2-ծառի տերևների քանակը չի գերազանցում $2^{T_s(n)}$ թիվը: Ուստի ստանում ենք, որ

$$n \le 2^{T_s(n)} \Rightarrow \log_2 n \le T_s(n) \Rightarrow \lceil \log_2 n \rceil \le T_s(n),$$

քանի որ $T_s(n)$ թիվը բնական է:

1.7. *Թեորեմ:* $T_s(n) = T_r(n) = \lceil \log_2 n \rceil$:

Ապացույց: Ցույց տանք, որ $T_r(n) \le \lceil \log_2 n \rceil$: Դրա համար բավական է կառուցել $\lceil \log_2 n \rceil$ բարդություն ունեցող ալգորիթմ, որը գնդիկների $\{1, 2, \dots, n\}$ բազմությունից գտնում է ռադիոակտիվը: Դիտարկենք այդ խնդիրը լուծող 1.2. ալգորիթմը: Նրա բարդությունը նշանակենք $T(n)$: Ալգորիթմի նկարագրից հետևում է, որ $T(n) \le 1 + T\left(\left\lceil \frac{n}{2} \right\rceil\right)$: Օգտվելով այս անհավասարությունից՝ ինդուկցիայի եղանակով ցույց տանք, որ $T(n) \le \lceil \log_2 n \rceil$, որից կհետևի $T_r(n) \le \lceil \log_2 n \rceil$ անհավասարությունը: Երբ $n = 1$, ապա $T(1) = 0 = \lceil \log_2 1 \rceil$: Ենթադրենք բոլոր $1 \le k < n$ բնական թվերի համար $T(k) \le \lceil \log_2 k \rceil$ պնդումը ճիշտ է: Այդ դեպքում ունենք, որ

$$T(n) \le 1 + T\left(\left\lceil \frac{n}{2} \right\rceil\right) \le 1 + \left\lceil \log_2 \left\lceil \frac{n}{2} \right\rceil \right\rceil = \left\lceil \log_2 2 \left\lceil \frac{n}{2} \right\rceil \right\rceil = \lceil \log_2 n \rceil:$$

Վերջին երկու հավասարությունները հեշտությամբ կարելի է ստուգել՝ առանձին-առանձին դիտարկելով $n = 2^k$ և $1 < 2^k < n < 2^{k+1}$ դեպքերը: Այժմ թեորեմի ապացույցը հետևում է ապացուցված լեմմաներից:

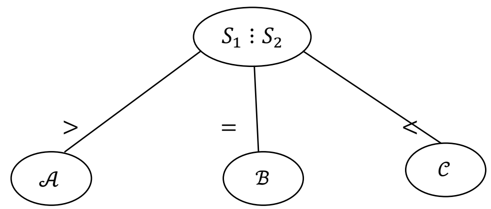
§ 1.2. ԿԵՂԾ ՄԵՏԱՂԱԴՐԱՄԻ ՈՐՈՆՈՒՄ

Դիցուք ունենք n հատ մետաղադրամ և հայտնի է, որ նրանցից որևէ մեկը կարող է լինել կեղծ՝ ծանր կամ թեթև մյուսներից: Լրացուցիչ տրված է մեկ իսկական մետաղադրամ: Ունենք նաև նժարավոր կշեռք, որի միջոցով կարող ենք համեմատել (բաղդատել) մետաղադրամների ցանկացած երկու խմբերի քաշերը և իմանալ՝ նրանք ունեն նույն քաշը, թե խմբերից մեկի քաշը մեծ է մյուս խմբի քաշից: Պահանջվում է նվազագույն թվով այդպիսի համեմատությունների (բաղդատումների) միջոցով գտնել կեղծ մետաղադրամը (եթե այդպիսին կա) և իմանալ՝ այն ծանր է, թե՞ թեթև իսկական մետաղադրամից:

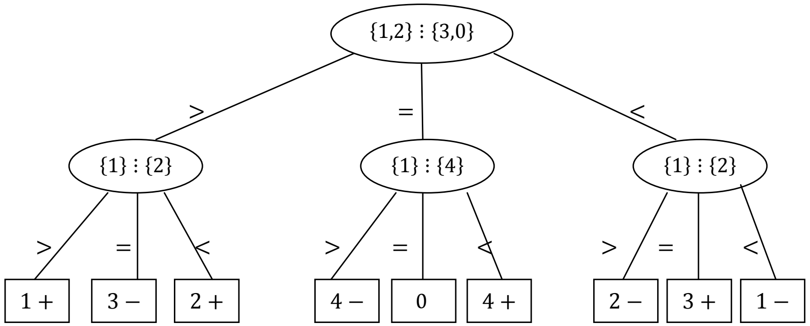
Կասկածելի մետաղադրամները համարակալենք $1, 2, \dots, n$ թվերով, իսկ իսկական մետաղադրամին վերագրենք 0 համարը: Յուրաքանչյուր $i = 1, 2, \dots, n$ համար i-րդ մետաղադրամի ծանր կամ թեթև լինելը համապատասխանաբար նշանակենք $i+$ և $i-$ սիմվոլներով: Նշանակենք $S \equiv \{0, 1, 2, \dots, n\}$: Խնդրի լուծումն ունի $2n + 1$ հատ հնարավոր դեպք՝ $1+, 2+, \dots, n+, 1-, 2-, \dots, n-$ և 0, որտեղ 0-ն այն դեպքն է, երբ բոլոր մետաղադրամներն իսկական են:

Այս խնդրի համար ևս լուծման ալգորիթմը կպատկերենք որոնման ծառի միջոցով: Քանի որ յուրաքանչյուր $S_1 \vdots S_2$ բաղդատում ($S_1, S_2 \subseteq S$) ունի երեք հնարավոր արդյունք ($S_1 > S_2, S_1 = S_2, S_1 < S_2$), ուստի այս դեպքում որոնման ծառը կլինի 3-ծառ: Որոնման ծառի արմատին համապատասխանեցնենք ալգորիթմի առաջին բաղդատումը: Եթե որոնման ծառի որևէ գագաթի համապատասխանեցված է $S_1 \vdots S_2$ բաղդատումը, ապա նրանից դուրս ելնող երեք կողերին վերագրենք $>$,$=$ և $<$ նշանները (նկար 1.4):

Նկար 1.4:



$\mathcal{A}$ գագաթին համապատասխանեցնենք այն բաղդատումը, որը կատարվում է ալգորիթմում $S_1 \vdots S_2$ համեմատությունից հետո, երբ $S_1 > S_2$: $\mathcal{B}$ գագաթին համապատասխանեցնենք ալգորիթմում կատարվող այն բաղդատումը, երբ $S_1 = S_2$, իսկ $\mathcal{C}$ գագաթին համապատասխանեցնենք ալգորիթմում կատարվող այն բաղդատումը, երբ $S_1 < S_2$: Ալգորիթմի աշխատանքի ավարտին համապատասխանող գագաթները համարենք որոնման ծառի տերևներ և նշենք ⊡ պայմանանշանով:



Նկար 1.5:

Հեշտ է ստուգել, որ 1.5 նկարում պատկերված որոնման ծառը հանդիսանում է խնդրի լուծման ալգորիթմ $n = 4$ դեպքում (եզրային գագաթներում նշված են համապատասխան արդյունքները):

Կասկածելի մետաղադրամների $\{1, 2, \dots, n\}$ բազմությունից կեղծ մետաղադրամը գտնող լավագույն ալգորիթմի բարդությունը նշանակենք $\tau(n)$, եթե հայտնի է, որ կա ամենաշատը մեկ իսկական մետաղադրամ: Լավագույն ալգորիթմի գոյությունն ակնհայտ է:

1.8. *Թեորեմ:* $\tau(n) = \lceil \log_3(2n+1) \rceil$:

Ապացույց: Լավագույն ալգորիթմին համապատասխան որոնման ծառը $\tau(n)$ բարձրություն ունեցող 3-ծառ է: Ուստի նրա տերևների քանակը չի գերազանցում $3^{\tau(n)}$ թիվը: Մյուս կողմից պարզ է, որ այդ ծառի տերևների քանակն առնվազն $(2n+1)$ է, քանի որ պետք է ունենա $1+, 2+, \dots, n+, 1-, 2-, \dots, n-$ և 0 դեպքերին համապատասխանող միմյանցից տարբեր տերևներ: Այնպես որ,

$$2n + 1 \le 3^{\tau(n)} \Rightarrow \log_3(2n+1) \le \tau(n) \Rightarrow \lceil \log_3(2n+1) \rceil \le \tau(n):$$

Ապացույցն ավարտելու համար բավական է ցույց տալ, որ երբ $2n + 1 \le 3^l$, ապա n մետաղադրամներից կեղծը, եթե այն կա, կարելի է գտնել ոչ շատ քան l կշռումների միջոցով:

Նախ քննարկենք այն դեպքը, երբ $2n + 1 = 3^l$ կամ որ նույնն է

$$n = \frac{1}{2}\left(3^l - 1\right) \equiv K_l:$$

Նշենք, որ $K_1 = 1, K_2 = 4$ և $K_l = 3K_{l-1} + 1$, երբ $l \ge 2$: Ստորև կտրվի K_l մետաղադրամներից l կշռումների միջոցով կեղծը գտնելու ալգորիթմի ընդհանուր նկարագիրը: Ալգորիթմի աշխատանքի ժամանակ կհանդիպեն հետևյալ երեք դեպքերը.

Առաջին դեպք: Մնացել է K_j $(1 \le j \le l)$ կասկածելի մետաղադրամ և ունենք առնվազն մեկ իսկական մետաղադրամ: Անհրաժեշտ է j կշռումների միջոցով գտնել կեղծ մետաղադրամը, եթե այն կա: Այս դեպքը համապատասխանում է նաև ալգորիթմի սկզբին:

Երկրորդ դեպք: Կեղծ մետաղադրամ կա. այն կա՛մ ծանր է և գտնվում է $K_j + 1$ կասկածելի ծանր մետաղադրամների մեջ, կա՛մ թեթև է և գտնվում է K_j կասկածելի թեթև մետաղադրամների մեջ: Անհրաժեշտ է j կշռումների միջոցով գտնել կեղծ մետաղադրամը $(1 \le j \le l)$:

Երրորդ դեպք: Կեղծ մետաղադրամ կա. այն կա՛մ թեթև է և գտնվում է $K_j + 1$ կասկածելի թեթև մետաղադրամների մեջ, կա՛մ ծանր է և գտնվում է K_j կասկածելի ծանր մետաղադրամների մեջ: Անհրաժեշտ է j կշռումների միջոցով գտնել կեղծ մետաղադրամը $(1 \le j \le l)$:

Կշռման եղանակն առաջին դեպքում: Կշեռքի A նժարի վրա դնում ենք $K_{j-1} + 1$ կասկածելի մետաղադրամ, իսկ B նժարի վրա՝ K_{j-1} կասկածելի և մեկ իսկական մետաղադրամ: Չի օգտագործվում K_{j-1} կասկածելի մետաղադրամ:

Կշռման արդյունքի վերլուծությունը.

ա) Բաղդրվող մետաղադրամները հավասար են $(A = B)$ և, հետևաբար, բոլոր բաղդրվող մետաղադրամներն իսկական են: Խնդրի լուծումն ավարտելու համար անհրաժեշտ է $(j-1)$ կշռումների միջոցով, չօգտագործված K_{j-1} կասկածելի մետաղադրամներից, գտնել կեղծ մետաղադրամը, եթե այն կա: Երբ $(j-1) > 1$, ապա խնդիրը բերվում է նույն առաջին դեպքին, իսկ երբ $(j-1) = 1$, ապա ունենք $K_1 = 1$

կասկածելի մետաղադրամ և մեկ կշռման միջոցով հարկավոր է որոշել, թե այն ինչպիսին է, որի լուծումն ակնհայտ է:

բ) $\boldsymbol{A > B}$: Կեղծ մետաղադրամ կա. այն կա՛մ ծանր է և գտնվում է $\boldsymbol{A}$ նժարի $\boldsymbol{K_{j-1}+1}$ կասկածելի ծանր մետաղադրամների մեջ, կա՛մ թեթև է և $\boldsymbol{B}$ նժարի $\boldsymbol{K_{j-1}}$ կասկածելի թեթև մետաղադրամներից մեկն է: Խնդրի լուծման համար անհրաժեշտ է $(\boldsymbol{j-1})$ կշռումների միջոցով գտնել կեղծ մետաղադրամը: Երբ $(\boldsymbol{j-1}) > \boldsymbol{1}$, ապա խնդիրը բերվում է նույն երկրորդ դեպքին, իսկ երբ $(\boldsymbol{j-1}) = \boldsymbol{1}$, այսինքն երբ կան երկու կասկածելի ծանր և մեկ կասկածելի թեթև մետաղադրամ, և անհրաժեշտ է մեկ բաղդատման միջոցով գտնել կեղծ մետաղադրամը, ապա խնդիրը լուծում է կասկածելի ծանր մետաղադրամների քաշերի համեմատությունը:

գ) $\boldsymbol{A < B}$: Կեղծ մետաղադրամ կա. այն կա՛մ թեթև է և գտնվում է $\boldsymbol{A}$ նժարի $\boldsymbol{K_{j-1}+1}$ կասկածելի թեթև մետաղադրամների մեջ, կա՛մ ծանր է և $\boldsymbol{B}$ նժարի $\boldsymbol{K_{j-1}}$ կասկածելի ծանր մետաղադրամներից մեկն է: Խնդրի լուծման համար անհրաժեշտ է $(\boldsymbol{j-1})$ կշռումների միջոցով գտնել կեղծ մետաղադրամը: Երբ $(\boldsymbol{j-1}) > \boldsymbol{1}$, ապա խնդիրը բերվում է նույն երրորդ դեպքին, իսկ $(\boldsymbol{j-1}) = \boldsymbol{1}$ դեպքում խնդիրը լուծում է երկու կասկածելի թեթև մետաղադրամների քաշերի համեմատությունը:

Կշռման եղանակն երկրորդ դեպքում: $\boldsymbol{A}$ և $\boldsymbol{B}$ նժարներից յուրաքանչյուրի վրա դնում ենք $\boldsymbol{K_{j-1}+1}$ կասկածելի ծանր և $\boldsymbol{K_{j-1}}$ կասկածելի թեթև մետաղադրամ և բաղդատում: Չի օգտագործվում $\boldsymbol{K_{j-1}}$ կասկածելի ծանր և $\boldsymbol{K_{j-1}+1}$ կասկածելի թեթև մետաղադրամ:

Կշռման արդյունքի վերլուծությունը.

ա) $\boldsymbol{A = B}$: Յուրաքանչյուր նժարի վրա գտնվող բոլոր մետաղադրամներն իսկական են: Չօգտագործված $\boldsymbol{K_{j-1}}$ կասկածելի ծանր և $\boldsymbol{K_{j-1}+1}$ կասկածելի թեթև մետաղադրամներից մեկը կեղծ է (եթե այդպիսին կա): Խնդիրը լուծելու համար հարկավոր է $(\boldsymbol{j-1})$ կշռումների միջոցով գտնել կեղծ մետաղադրամը: Երբ $(\boldsymbol{j-1}) > \boldsymbol{1}$, ապա խնդիրը բերվում է երրորդ դեպքին, իսկ $(\boldsymbol{j-1}) = \boldsymbol{1}$ դեպքում՝ լուծումն ակնհայտ է:

բ) $\boldsymbol{A > B}$: Կեղծ մետաղադրամ կա. այն կա՛մ ծանր է և գտնվում է $\boldsymbol{A}$ նժարի $\boldsymbol{K_{j-1}+1}$ կասկածելի ծանր մետաղադրամների մեջ, կա՛մ թեթև է և $\boldsymbol{B}$ նժարի $\boldsymbol{K_{j-1}}$ կասկածելի թեթև մետաղադրամներից մեկն է: Անհրաժեշտ է $(\boldsymbol{j-1})$ կշռումների միջոցով գտնել կեղծ մետաղադրամը՝ ունենալով

առնվազն մեկ իսկական մետաղադրամ: Հանգեցինք նույն երկրորդ դեպքի խնդրին $(\boldsymbol{j-1})$ արժեքի համար:

գ) $\boldsymbol{A < B}$: Արդյունքը նույնպես բերվում է երկրորդ դեպքի խնդրին $(\boldsymbol{j-1})$ արժեքի համար:

Կշռման եղանակն երրորդ դեպքում: $\boldsymbol{A}$ և $\boldsymbol{B}$ նժարներից յուրաքանչյուրի վրա դնում ենք $\boldsymbol{K_{j-1}+1}$ կասկածելի թեթև և $\boldsymbol{K_{j-1}}$ կասկածելի ծանր մետաղադրամ և բաղդատում: Չի օգտագործվում $\boldsymbol{K_{j-1}}$ կասկածելի թեթև և $\boldsymbol{K_{j-1}+1}$ կասկածելի ծանր մետաղադրամ:

Կշռման արդյունքի վերլուծությունը.

ա) $\boldsymbol{A = B}$: Յուրաքանչյուր նժարի վրա գտնվող բոլոր մետաղադրամներն իսկական են: Չօգտագործված $\boldsymbol{K_{j-1}}$ կասկածելի թեթև և $\boldsymbol{K_{j-1}+1}$ կասկածելի ծանր մետաղադրամներից մեկը կեղծ է (եթե այդպիսին կա): Խնդիրը լուծելու համար հարկավոր է $(\boldsymbol{j-1})$ կշռումների միջոցով գտնել կեղծ մետաղադրամը: Երբ $(\boldsymbol{j-1}) > \boldsymbol{1}$, ապա խնդիրը բերվում է երկրորդ դեպքին, իսկ $(\boldsymbol{j-1}) = \boldsymbol{1}$ դեպքում՝ լուծումն ակնհայտ է:

բ) $\boldsymbol{A > B}$: Կեղծ մետաղադրամ կա. այն կա՛մ ծանր է և գտնվում է $\boldsymbol{A}$ նժարի $\boldsymbol{K_{j-1}}$ կասկածելի ծանր մետաղադրամների մեջ, կա՛մ թեթև է և $\boldsymbol{B}$ նժարի $\boldsymbol{K_{j-1}+1}$ կասկածելի թեթև մետաղադրամներից մեկն է: Անհրաժեշտ է $(\boldsymbol{j-1})$ կշռումների միջոցով գտնել կեղծ մետաղադրամը՝ ունենալով առնվազն մեկ իսկական մետաղադրամ: Հանգեցինք նույն երրորդ դեպքի խնդրին $(\boldsymbol{j-1})$ արժեքի համար:

գ) $\boldsymbol{A < B}$: Արդյունքը նմանապես բերվում է երրորդ դեպքի խնդրին $(\boldsymbol{j-1})$ արժեքի համար:

Այժմ ինդուկցիայի եղանակով ապացուցենք, որ ունենալով առնվազն մեկ իսկական մետաղադրամ՝ $\boldsymbol{K_l = \frac{1}{2}(3^l - 1)}$ կասկածելի մետաղադրամներից կեղծը (եթե այն կա) կարելի է գտնել $\boldsymbol{l}$ կշռումների միջոցով: Երբ $\boldsymbol{l = 1}$, ապա առաջին, երկրորդ և երրորդ դեպքերին համապատասխան խնդիրների լուծումներն ակնհայտ են և նկարագրված են առաջին կշռման արդյունքների վերլուծություններում: Ենթադրենք $\boldsymbol{l} > 1$ և առաջին, երկրորդ և երրորդ դեպքերին համապատասխան խնդիրները $(\boldsymbol{l-1})$ արժեքի համար կարելի է լուծել $(\boldsymbol{l-1})$ կշռումների միջոցով: Վերևում ցույց է տրված, որ առաջին, երկրորդ և երրորդ դեպքերին համապատասխան խնդիրները $\boldsymbol{l}$ արժեքի համար բերվում են այդ դեպքերից մեկին համապատասխանող խնդրին $(\boldsymbol{l-1})$ արժեքի համար՝

կատարելով ճիշտ մեկ կշռում: Ուստի, համաձայն ինդուկցիայի ենթադրության, բոլոր դեպքերին համապատասխան խնդիրները $\boldsymbol{l}$ արժեքի համար կարելի է լուծել $\boldsymbol{l}$ կշռումների միջոցով:

Նկարագրենք $\boldsymbol{K_l}$ կասկածելի մետաղադրամներից $\boldsymbol{l}$ կշռումների միջոցով կեղծ մետաղադրամը (եթե այն կա) գտնելու՝ **Կարեն Ազատյանի** առաջարկած ալգորիթմը՝ ունենալով առնվազն մեկ իսկական մետաղադրամ: Խնդրի լուծումն ակնհայտ է, երբ $\boldsymbol{l = 1}$: Ենթադրենք $\boldsymbol{l > 1}$ և $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամներից կեղծը (եթե այն կա) կարելի է գտնել $(\boldsymbol{l - 1})$ կշռումների միջոցով:

Կշռման եղանակը: Կշեռքի $\boldsymbol{A}$ նժարի վրա տեղադրենք $\boldsymbol{K_{l-1} + 1}$ կասկածելի մետաղադրամ, իսկ $\boldsymbol{B}$ նժարի վրա՝ $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամ և մեկ իսկական մետաղադրամ: Չի օգտագործվում $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամ:

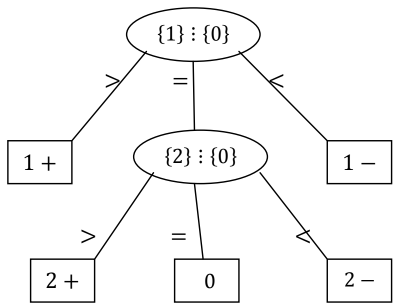
Կշռման արդյունքի վերլուծությունը.

ա) $\boldsymbol{A = B}$: Այս դեպքում համեմատվող մետաղադրամներն իսկական են: Չօգտագործված $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամներից կեղծը (եթե այն կա), համաձայն ինդուկցիայի ենթադրության, կարելի է գտնել $(\boldsymbol{l - 1})$ կշռումների միջոցով:

բ) $\boldsymbol{A > B}$: Կեղծ մետաղադրամ կա. այն կա՛մ ծանր է և գտնվում է $\boldsymbol{A}$ նժարի $\boldsymbol{K_{l-1} + 1}$ կասկածելի ծանր մետաղադրամների մեջ, կա՛մ թեթև է և $\boldsymbol{B}$ նժարի $\boldsymbol{K_{l-1}}$ կասկածելի թեթև մետաղադրամներից մեկն է: Վերցնելով մեկ կասկածելի թեթև մետաղադրամ $\boldsymbol{B}$ նժարից և մեկ կասկածելի ծանր մետաղադրամ $\boldsymbol{A}$ նժարից՝ կազմենք կասկածելի թեթև և կասկածելի ծանր մետաղադրամների զույգ, որը կդիտարկենք որպես մեկ կասկածելի նոր մետաղադրամ: Կասկածելի նոր մետաղադրամների քանակը հավասար է $\boldsymbol{K_{l-1}}$: Մեզ տրված մեկ իսկական մետաղադրամը և չօգտագործված $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամներից (որոնք իրականում իսկական են) մեկը կկազմեն իսկական նոր մետաղադրամ: Զույգեր կազմելիս չի օգտագործվել մեկ կասկածելի ծանր մետաղադրամ $\boldsymbol{A}$ նժարից: Ունենալով մեկ նոր իսկական մետաղադրամ և $\boldsymbol{K_{l-1}}$ կասկածելի նոր մետաղադրամ, համաձայն ինդուկցիայի ենթադրության, $(\boldsymbol{l - 1})$ կշռումների միջոցով կարելի է գտնել կեղծ նոր մետաղադրամը, եթե այն կա: Եթե կեղծ նոր մետաղադրամ չկա, ապա կեղծ մետաղադրամը չօգտագործված մեկ կասկածելի ծանր մետաղադրամն է $\boldsymbol{A}$ նժարից: Եթե կեղծ նոր մետաղադրամ կա և այն ծանր (թեթև) է նոր իսկական մետաղադրամից, ապա այդ կեղծ նոր մետաղադրամի մեջ մտնող կասկածելի ծանր (թեթև) մետաղադրամն որոնելի կեղծ մետաղադրամն է:

գ) $\boldsymbol{A < B}$: Կեղծ մետաղադրամ կա. այն կա՛մ թեթև է և գտնվում է $\boldsymbol{A}$ նժարի $\boldsymbol{K_{j-1} + 1}$ կասկածելի թեթև մետաղադրամների մեջ, կա՛մ ծանր է և $\boldsymbol{B}$ նժարի $\boldsymbol{K_{j-1}}$ կասկածելի ծանր մետաղադրամներից մեկն է: Վերցնելով մեկ կասկածելի ծանր մետաղադրամ $\boldsymbol{B}$ նժարից և մեկ կասկածելի թեթև մետաղադրամ $\boldsymbol{A}$ նժարից՝ կազմենք կասկածելի ծանր և կասկածելի թեթև մետաղադրամների զույգ, որը կդիտարկենք որպես մեկ կասկածելի նոր մետաղադրամ: Կասկածելի նոր մետաղադրամների քանակը հավասար է $\boldsymbol{K_{l-1}}$: Մեզ տրված մեկ իսկական մետաղադրամը և չօգտագործված $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամներից (որոնք իրականում իսկական են) մեկը կկազմեն իսկական նոր մետաղադրամ: Զույգեր կազմելիս չի օգտագործվել մեկ կասկածելի թեթև մետաղադրամ $\boldsymbol{A}$ նժարից: Ունենալով մեկ նոր իսկական մետաղադրամ և $\boldsymbol{K_{l-1}}$ կասկածելի նոր մետաղադրամ, համաձայն ինդուկցիայի ենթադրության, $(\boldsymbol{l - 1})$ կշռումների միջոցով կարելի է գտնել կեղծ նոր մետաղադրամը, եթե այն կա: Եթե կեղծ նոր մետաղադրամ չկա, ապա կեղծ մետաղադրամը չօգտագործված մեկ կասկածելի թեթև մետաղադրամն է $\boldsymbol{A}$ նժարից: Եթե կեղծ նոր մետաղադրամ կա և այն ծանր (թեթև) է նոր իսկական մետաղադրամից, ապա այդ կեղծ նոր մետաղադրամի մեջ մտնող կասկածելի ծանր (թեթև) մետաղադրամն որոնելի կեղծ մետաղադրամն է:

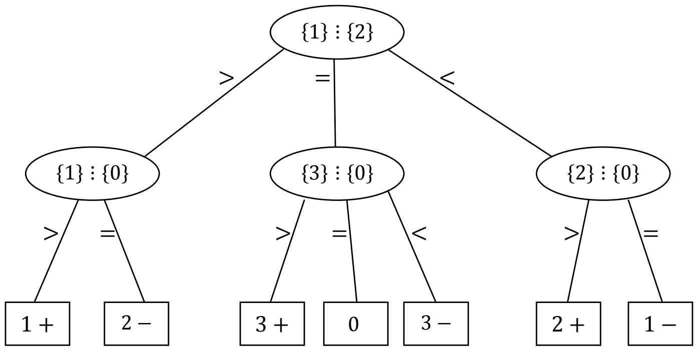
Այսպիսով, երեք դեպքերում էլ որոնելի կեղծ մետաղադրամը (եթե այն կա), կարելի է գտնել $(\boldsymbol{l - 1})$ կշռումների միջոցով: Ուստի $\boldsymbol{K_l}$ կասկածելի մետաղադրամներից հնարավոր կեղծը կարելի է գտնել $\boldsymbol{l}$ կշռումների՝ ունենալով առնվազն մեկ իսկական մետաղադրամ:



Նկար 1.6:

Այժմ վերադառնանք կեղծ մետաղադրամի որոնման խնդրին ընդհանուր դեպքում, երբ ունենք մեկ իսկական մետաղադրամ և $\mathbf{1, 2, \ldots, n}$ կասկածելի մետաղադրամ:

Պարզ է, որ $\boldsymbol{n = 2}$ կամ $\boldsymbol{n = 3}$ դեպքում $\boldsymbol{K_1 < n < K_2}$ և խնդիրը լուծվում է երկու կշռումների միջոցով (տես. 1.6 և 1.7 նկարները): Հաջորդիվ ենթադրենք, որ $\boldsymbol{K_{l-1} < n < K_l}$ (որտեղ $\boldsymbol{l \geq 3}$) և բոլոր $\boldsymbol{m}$ արժեքների համար, որոնք բավարարում են $\boldsymbol{1 \leq m \leq K_{l-1}}$ անհավասարություններին, $\boldsymbol{m}$ մետաղադրամներից կեղծը կարելի է գտնել ամենաշատը $\boldsymbol{(l-1)}$ կշռումների միջոցով: Այդ դեպքում $\boldsymbol{n}$ կասկածելի մետաղադրամների բազմությունից առանձնացնում ենք $\boldsymbol{K_{l-1}}$ մետաղադրամ, մնացածները բաժանում ենք երկու հավասար մասերի (եթե կենտ են, ապա ավելացնում ենք իսկական մետաղադրամը) և բաղդատում իրար հետ: $\boldsymbol{A}$ և $\boldsymbol{B}$ նժարներից յուրաքանչյուրի վրա գտնվում է ամենաշատը $\left\lceil\frac{n-K_{l-1}}{2}\right\rceil$ կասկածելի մետաղադրամ:



Նկար 1.7:

Եթե $\boldsymbol{A = B}$, ապա նժարների վրա գտնվող մետաղադրամներն իսկական են և մնացած $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամներից կեղծ մետաղադրամը (եթե այդպիսին կա) կորոշենք $\boldsymbol{(l-1)}$ կշռումների միջոցով: Եթե $\boldsymbol{A > B}$ ($\boldsymbol{A < B}$), ապա կեղծ մետաղադրամ կա. այն կա՛մ ծանր (թեթև) է և $\boldsymbol{A}$ նժարի $\left\lceil\frac{n-K_{l-1}}{2}\right\rceil$ կասկածելի ծանր (թեթև) մետաղադրամներից մեկն է, կա՛մ թեթև (ծանր) է և $\boldsymbol{B}$ նժարի $\left\lceil\frac{n-K_{l-1}}{2}\right\rceil$ կասկածելի թեթև (ծանր) մետաղադրամներից մեկն է, երբ $\boldsymbol{n - K_{l-1} \equiv 0(mod\ 2)}$ ($\boldsymbol{n - K_{l-1} \equiv 1(mod\ 2)}$ դեպքում՝ $\boldsymbol{B}$ նժարի $\left\lceil\frac{n-K_{l-1}}{2}\right\rceil - \mathbf{1}$ կասկածելի թեթև (ծանր) մետաղադրամներից մեկն է): Վերցնելով մեկ կասկածելի թեթև (ծանր) մետաղադրամ $\boldsymbol{B}$ նժարից և մեկ կասկածելի ծանր (թեթև) մետաղադրամ $\boldsymbol{A}$ նժարից՝ կազմենք կասկածելի թեթև (ծանր) և կասկածելի ծանր (թեթև) մետաղադրամների

զույգ, որը կդիտարկենք որպես մեկ կասկածելի նոր մետաղադրամ: Կասկածելի նոր մետաղադրամների քանակը հավասար է $\left\lceil\frac{n-K_{l-1}}{2}\right\rceil$, երբ $\boldsymbol{n - K_{l-1} \equiv 0(mod\ 2)}$, և $\left\lceil\frac{n-K_{l-1}}{2}\right\rceil - \mathbf{1}$, երբ $\boldsymbol{n - K_{l-1} \equiv 1(mod\ 2)}$: Մեզ տրված մեկ իսկական մետաղադրամը և չօգտագործված $\boldsymbol{K_{l-1}}$ կասկածելի մետաղադրամներից (որոնք իրականում իսկական են) մեկը կկազմեն իսկական նոր մետաղադրամ: Զույգեր կազմելիս, $\boldsymbol{n - K_{l-1} \equiv 1(mod\ 2)}$ դեպքում, չի օգտագործվում մեկ կասկածելի ծանր (թեթև) մետաղադրամ: Ունենք որ $\boldsymbol{l \geq 3}$ և $\boldsymbol{K_{l-1} < n < K_l}$: Հետևաբար ճշմարիտ են հետևյալ անհավասարությունների շարքը.

$$\boldsymbol{K_{l-1} < n < K_l \Rightarrow 0 < n - K_{l-1} < K_l - K_{l-1} = 3^{l-1} \Rightarrow}$$

$$\boldsymbol{1 \leq n - K_{l-1} \leq 3^{l-1} - 1 \Rightarrow \frac{1}{2} \leq \frac{n - K_{l-1}}{2} \leq \frac{3^{l-1} - 1}{2} = K_{l-1} \Rightarrow}$$

$$\boldsymbol{1 \leq \left\lceil\frac{n - K_{l-1}}{2}\right\rceil \leq K_{l-1}}:$$

Ունենալով մեկ նոր իսկական մետաղադրամ և ամենաշատը $\left\lceil\frac{n-K_{l-1}}{2}\right\rceil$ կասկածելի նոր մետաղադրամ, համաձայն վերևում արված ենթադրության, $\boldsymbol{(l-1)}$ կշռումների միջոցով կարելի է գտնել կեղծ նոր մետաղադրամը ($\boldsymbol{n - K_{l-1} \equiv 0(mod\ 2)}$ դեպքում կեղծ նոր մետաղադրամ անպայման գոյություն ունի, իսկ $\boldsymbol{n - K_{l-1} \equiv 1(mod\ 2)}$ դեպքում նրա գոյությունը պարտադիր չէ): Եթե $\boldsymbol{n - K_{l-1} \equiv 1(mod\ 2)}$ դեպքում կեղծ նոր մետաղադրամ չկա, ապա որոնելի կեղծ մետաղադրամը չօգտագործված մեկ կասկածելի ծանր (թեթև) մետաղադրամն է: Եթե կեղծ նոր մետաղադրամ կա և այն ծանր (թեթև) է, ապա այդ կեղծ նոր մետաղադրամի մեջ մտնող կասկածելի ծանր (թեթև) մետաղադրամն որոնելի կեղծ մետաղադրամն է:

Թեորեմի ապացույցն ամբողջությամբ ավարտված է:

Այժմ ապացուցենք, որ $\boldsymbol{n = K_l = \frac{3^l - 1}{2}}$ դեպքում իսկական մետաղադրամ ունենալն անհարժեշտ է $\boldsymbol{l}$ կշռումների միջոցով խնդիրը լուծելու համար: Ենթադրենք տրված չէ իսկական մետաղադրամ, բայց գոյություն ունի խնդիրը լուծող $\boldsymbol{l}$ բարդությամբ ալգորիթմ: Դիցուք ալգորիթմի առաջին քայլում միմյանց հետ համեմատվում են $\boldsymbol{x}$ թվով մետաղադրամներ (յուրաքանչյուր նժարի վրա): Հեշտ է նկատել, որ

a) Համեմատվող մետաղադրամների քաշերը միմյանց հավասար լինելու դեպքում խնդրի լուծումը $\boldsymbol{(l-1)}$ քայլից գտնելու համար պետք է բավարարվի $K_l - 2x \le K_{l-1}$ կամ, որ նույնն է, ինչ $2x \ge K_l - K_{l-1} = 3^{l-1}$ անհավասարությունը.

b) Համեմատվող մետաղադրամների քաշերը տարբեր լինելու դեպքում կա կեղծ մետաղադրամ, իսկ հնարավոր դեպքերի քանակը հավասար է $2x$ և $(l-1)$ քայլերի միջոցով խնդրի լուծումը գտնելու համար պետք է տեղի ունենա $2x \le 3^{l-1}$ անհավասարությունը:

Պարզ է, որ գոյություն չունի այդ անհավասարություններին բավարարող x ամբողջ թիվ:

Առաջարկվում է ապացուցել, որ $n = K_l - 1$ դեպքում, երբ տրված չէ իսկական մետաղադրամ, խնդիրը կարելի է լուծել l կշռումների միջոցով:

§ 1.3. ԵՐԿԸՆԹԱՑ ՀԱՋՈՐԴԱԿԱՆՈՒԹՅԱՆ ԱՄԵՆԱՄԵԾ ՏԱՐՐԻ ՈՐՈՆՈՒՄ

1.9. ***Սահմանում:*** *Դրական թվերի* $a_1, a_2, \dots, a_n$ *վերջավոր հաջորդականությունը կոչվում է* ***երկընթաց****, եթե տեղի ունի հետևյալ երեք պայմաններից ճիշտ մեկը.*

1) $0 < a_1 < a_2 < \dots < a_n$;
2) $a_1 > a_2 > \dots > a_n > 0$;
3) *Գոյություն ունի այնպիսի* k $(1 < k < n)$ *բնական թիվ, որ*

$$0 < a_1 < a_2 < \dots < a_k \text{ և } a_k > a_{k+1} > \dots > a_n > 0,$$

այսինքն՝ մինչև k*-րդ անդամը հաջորդականությունն աճում է, իսկ հետո՝ նվազում:*

n անդամ պարունակող բոլոր երկընթաց հաջորդականությունների բազմությունը նշանակենք Z_n սիմվոլով:

Դիտարկենք հետևյալ խնդիրը. Տրված է մեզ անհայտ $\alpha = (x_1, x_2, \dots, x_n) \in Z_n$ երկընթաց հաջորդականությունը: Թույլատրվում է հարցնել, թե ինչի է հավասար հաջորդականության i-րդ $(1 \le i \le n)$ անդամի արժեքը և ստանալ այդ հարցի պատասխանը: Պահանջվում է նվազագույն թվով հարցերի միջոցով գտնել α հաջորդականության ամենամեծ անդամը: Խնդիրը լուծող լավագույն ալգորիթմի գոյությունն ակնհայտ է:

Ցանկացած k բնական թվի համար λ_k սիմվոլով նշանակենք այն ամենամեծ ամբողջ թիվը, որը բավարարում է հետևյալ պայմանին. $\alpha = (x_1, x_2, \dots, x_{\lambda_k}) \in Z_{\lambda_k}$ երկընթաց հաջորդականության ամենամեծ անդամը կարելի է գտնել k հարցերի միջոցով: Պարզ է, որ ցանկացած k բնական թվի համար λ_k թիվ գոյություն ունի և $\lambda_1 = 1$, $\lambda_2 = 2$:

1.10. *Լեմմա: Ցանկացած* $k \ge 3$ *բնական թվի համար տեղի ունի*

$$\lambda_k \le \lambda_{k-1} + \lambda_{k-2} + 1$$

անհավասարությունը:

Ապացույց: Դիցուք $k \ge 3$ բնական թիվ է, և $\mathcal{A}$ հանդիսանում է երկընթաց հաջորդականության ամենամեծ տարրն որոնող լավագույն ալգորիթմ: Ենթադրենք $\alpha = (x_1, x_2, \dots, x_{\lambda_k}) \in Z_{\lambda_k}$ երկընթաց հաջորդականության համար $\mathcal{A}$ ալգորիթմի առաջին և երկրորդ հարցերով որոշվում են նրա՝ i_1 և i_2 համարներն ունեցող անդամների արժեքները, որտեղ $i_1 < i_2$:

Նկատենք, որ $i_1 - 1 \le \lambda_{k-2}$: Իսկապես, եթե $i_1 - 1 > \lambda_{k-2}$, ապա $\mathcal{A}$ ալգորիթմը չի կարող մնացած $k-2$ հարցերի միջոցով որոշել $\alpha = (x_1, x_2, \dots, x_{\lambda_k}) \in Z_{\lambda_k}$ երկընթաց հաջորդականության ամենամեծ անդամը, երբ այն պատկանում է $\{x_1, x_2, \dots, x_{i_1-1}\}$ բազմությանը:

Մյուս կողմից, եթե $\alpha = (x_1, x_2, \dots, x_{\lambda_k}) \in Z_{\lambda_k}$ հաջորդականության ամենամեծ անդամը պատկանում է $\{x_{i_1+1}, \dots, x_{i_2-1}, x_{i_2}, x_{i_2+1}, \dots, x_{\lambda_k}\}$ բազմությանը, ապա $\lambda_k - i_1 \le \lambda_{k-1}$, քանի որ մեկ հարցի պատասխանը՝ x_{i_2} անդամի արժեքն արդեն ունենք և լրացուցիչ հարցերի քանակը $\le (k-2)$:

Գումարելով ստացված երկու անհավասարությունները՝ կստանանք, որ

$$\lambda_k - 1 \le \lambda_{k-1} + \lambda_{k-2} \ \Rightarrow\ \lambda_k \le \lambda_{k-1} + \lambda_{k-2} + 1:$$

Լեմման ապացուցված է:

1.11. ***Լեմմա:*** *Երբ* $\phi_0 = \phi_1 = 1$ *և* $2 \le k \in \mathbb{Z}$, *ապա* $\phi_k = \phi_{k-1} + \phi_{k-2}$ *անդրադարձ առնչությանը բավարարող հաջորդականության ընդհանուր անդամը տրվում է*

$$\phi_k = \frac{1}{\sqrt{5}}\left[\left(\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1}{2}-\frac{\sqrt{5}}{2}\right)^{k+1}\right]$$

բանաձևով, որտեղ $0 \le k \in \mathbb{Z}$: *Այս հաջորդականությունը կոչվում է* ***Ֆիբոնաչիի հաջորդականություն:***

Ապացույց: Նախ պարզ է, որ նշված առնչությանը բավարարող հաջորդականությունը միակն է: Հեշտ է ստուգել, որ եթե γ_1 և γ_2 թվերը հանդիսանում են $x^2 = x + 1$ հավասարման լուծումներ, ապա $\gamma_i, {\gamma_i}^2, {\gamma_i}^3, \dots, {\gamma_i}^n, \dots$ հաջորդականությունը $(i = 1, 2)$ բավարարում է ${\gamma_i}^k = {\gamma_i}^{k-1} + {\gamma_i}^{k-2}$ առնչությանը: Հետևաբար $\phi_k = \phi_{k-1} + \phi_{k-2}$ անդրադարձ առնչության լուծումն ունի $\phi_k = c_1{\gamma_1}^k + c_2{\gamma_2}^k$ տեսքը, որտեղ c_1 և c_2 թվերը բավարարում են

$$\begin{cases} c_1 + c_2 = 1 \\ c_1\gamma_1 + c_2\gamma_2 = 1 \end{cases}$$

հավասարումների համակարգին: Հաշվի առնելով, որ $\gamma_1 = \frac{1}{2} + \frac{\sqrt{5}}{2}$, $\gamma_2 = \frac{1}{2} - \frac{\sqrt{5}}{2}$, և լուծելով վերջին համակարգը՝ ստանում ենք, որ

$$c_1 = \frac{\gamma_2-1}{\gamma_2-\gamma_1} = \frac{1}{\sqrt{5}}\left(\frac{1}{2}+\frac{\sqrt{5}}{2}\right) \text{ և } c_1 = \frac{\gamma_1-1}{\gamma_1-\gamma_2} = -\frac{1}{\sqrt{5}}\left(\frac{1}{2}-\frac{\sqrt{5}}{2}\right):$$

Հետևաբար $\phi_k = \frac{1}{\sqrt{5}}\left[\left(\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1}{2}-\frac{\sqrt{5}}{2}\right)^{k+1}\right]$, երբ $0 \le k \in \mathbb{Z}$:

1.12. ***Լեմմա:*** *Ցանկացած k բնական թվի համար տեղի ունի*

$$\lambda_k \le \phi_{k+1} - 1$$

անհավասարությունը:

Ապացույց: $k = 1$ և $k = 2$ դեպքերում պնդումն ակնհայտ է.

$$1 = \lambda_1 \le \phi_2 - 1 = 2 - 1 = 1,$$

$$2 = \lambda_2 \le \phi_3 - 1 = 3 - 1 = 2:$$

Ենթադրենք $k > 2$ և k թվից փոքր թվերի համար լեմման ճիշտ է, այսինքն՝

$$\lambda_{k-1} \le \phi_k - 1 \text{ և } \lambda_{k-2} \le \phi_{k-1} - 1:$$

Այդ դեպքում, ըստ 1.10 լեմմայի, ստանում ենք, որ

$$\lambda_k \le \lambda_{k-1} + \lambda_{k-2} + 1 \le \phi_k - 1 + \phi_{k-1} - 1 + 1 = \phi_k + \phi_{k-1} - 1 = \phi_{k+1} - 1$$

և լեմման ապացուցված է:

1.13. ***Լեմմա:*** *Եթե $k \ge 3$ բնական թիվ է և հայտնի է $\alpha = (x_1, x_2, \dots, x_{\phi_{k-1}}) \in Z_{\phi_{k-1}}$ երկընթաց հաջորդականության $x_{\phi_{k-1}}$ անդամի արժեքը, ապա այդ հաջորդականության ամենամեծ անդամը կարելի է գտնել $k-2$ հարցերի միջոցով:*

Ապացույց: Երբ $k = 3$, ապա

$$\phi_3 - 1 = 2,\ x_{\phi_{3-1}} = x_{\phi_2} = x_2 \text{ և } k - 2 = 1,$$

որը նշանակում է, որ հայտնի է (x_1, x_2) երկընթաց հաջորդականության x_2 անդամի արժեքը և հարկավոր է մեկ հարցի միջոցով գտնել (x_1, x_2) հաջորդականության ամենամեծ անդամի արժեքը: Ակնհայտ է, որ x_1 անդամի

արժեքն իմանալով՝ կորոշենք (x_1, x_2) հաջորդականության ամենամեծ անդամի արժեքը:

Ենթադրենք լեմման ճշմարիտ է $k = p$ դեպքում և այն ապացուցենք $k = p + 1$ դեպքի համար:

Դիցուք ունենք $x_1, \dots, x_{\phi_p - 1}, x_{\phi_p}, x_{\phi_p + 1}, \dots, x_{\phi_{p+1} - 1}$ երկընթաց հաջորդականությունը և հայտնի է նրա x_{ϕ_p} անդամի արժեքը: Մեկ հարցի միջոցով որոշենք $x_{\phi_{p-1}}$ անդամի արժեքը: Հնարավոր են հետևյալ դեպքերը.

a) $x_{\phi_{p-1}} > x_{\phi_p}$: Պարզ է, որ $x_1, x_2, \dots, x_{\phi_{p+1}-1}$ հաջորդականության ամենամեծ անդամը հարկավոր է փնտրել $x_1, x_2, \dots, x_{\phi_p - 1}$ ենթահաջորդականության անդամների մեջ, որոնցից հայտնի է $x_{\phi_{p-1}}$ անդամի արժեքը: Համաձայն ինդուկցիայի ենթադրության, դա կարելի է անել $p - 2$ հարցերի միջոցով:

b) $x_{\phi_{p-1}} < x_{\phi_p}$: Այս դեպքում $x_1, x_2, \dots, x_{\phi_{p+1}-1}$ հաջորդականության ամենամեծ անդամը պետք է փնտրել $x_{\phi_{p-1}+1}, x_{\phi_{p-1}+2}, \dots, x_{\phi_{p+1}-1}$ ենթահաջորդականության անդամների մեջ: Նկատենք, որ եթե դիտարկենք այդ ենթահաջորդականության շրջված՝ $y_1 = x_{\phi_{p+1}-1}, y_2 = x_{\phi_{p+1}-2}, \dots, y_t = x_{\phi_{p-1}+1}$ հաջորդականությունը, ապա այն կլինի երկընթաց, որտեղ
$$\phi_{p+1} - t = \phi_{p-1} + 1 \Rightarrow t = \phi_{p+1} - \phi_{p-1} - 1 = \phi_p - 1,$$
և հայտնի է $y_{\phi_{p-1}} = x_{\phi_{p+1} - \phi_{p-1}} = x_{\phi_p}$ անդամի արժեքը: Համաձայն ինդուկցիայի ենթադրության, $y_1, y_2, \dots, y_{\phi_p - 1}$ երկընթաց հաջորդականության ամենամեծ անդամի արժեքը կարելի է գտնել $p - 2$ հարցերի միջոցով, երբ հայտնի է $y_{\phi_{p-1}}$ անդամի արժեքը:

Լեմման ապացուցված է ամբողջությամբ:

Պայմանավորվենք n անդամ ունեցող երկընթաց հաջորդականության ամենամեծ անդամն որոնող լավագույն ալգորիթմի բարդությունը նշանակել $t(n)$:

1.14. ***Թեորեմ:*** *Երբ $k \ge 3$ և $\phi_k \le n < \phi_{k+1}$, ապա $t(n) = k$:*

Ապացույց: Համաձայն 1.12. լեմմայի, ունենք, որ $n \ge \phi_k > \phi_k - 1 \ge \lambda_{k-1}$: Հետևաբար $t(n) \ge k$:

Մյուս կողմից, քանի որ $n < \phi_{k+1}$, ապա թեորեմի ապացույցն ավարտելու համար բավական է ցույց տալ, որ եթե $n = \phi_{k+1} - 1$, ապա Z_n բազմության յուրաքանչյուր երկընթաց հաջորդականության ամենամեծ անդամի արժեքը կարելի է որոշել k հարցերի միջոցով:

Դիցուք ունենք $\alpha = (x_1, x_2, \dots, x_{\phi_{k+1}-1})$ երկընթաց հաջորդականությունը: Առաջին երկու հարցերի միջոցով որոշենք $x_{\phi_{k-1}}$ և x_{ϕ_k} անդամների արժեքները:

Եթե $x_{\phi_{k-1}} > x_{\phi_k}$, ապա α հաջորդականության ամենամեծ անդամը գտնվում է $x_1, x_2, \dots, x_{\phi_k-1}$ ենթահաջորդականության անդամների մեջ: Այն բավարարում է 1.13. լեմմայի պայմաններին, ուստի $k-2$ հարցերի միջոցով կարելի է գտնել նրա ամենամեծ անդամի արժեքը:

Մյուս՝ $x_{\phi_{k-1}} < x_{\phi_k}$, դեպքում α հաջորդականության ամենամեծ անդամը գտնվում է $x_{\phi_{k-1}+1}, x_{\phi_{k-1}+2}, \dots, x_{\phi_{k+1}-1}$ ենթահաջորդականության անդամների մեջ: Նկատենք, որ այս ենթահաջորդականության շրջված՝ $y_1 = x_{\phi_{k+1}-1}, y_2 = x_{\phi_{k+1}-2}, \dots, y_{\phi_k-1} = x_{\phi_{k-1}+1}$ հաջորդականությունը բավարարում է 1.13. լեմմայի պայմաններին, ուստի $k-2$ հարցերի միջոցով կարելի է գտնել նրա ամենամեծ անդամի արժեքը:

Այսպիսով, երկու դեպքում էլ $k-2$ հարցերի միջոցով կարելի է գտնել α հաջորդականության ամենամեծ անդամի արժեքը: Թեորեմն ապացուցված է:

§ 1.4. ԲԱԶՄՈՒԹՅՈՒՆՆԵՐԻ ՀԱՎԱՍԱՐՈՒԹՅԱՆ ՍՏՈՒԳՈՒՄ

Դիցուք ունենք $A = \{a_1, a_2, \dots, a_n\}$ և $B = \{b_1, b_2, \dots, b_n\}$ բազմությունները: Ցանկացած $a_i \in A$ և $b_j \in B$ տարրերի համար կարող ենք ստանալ $a_i = b_j$ հարցի պատասխանը: Այդ դեպքում պահանջվում է նվազագույն թվով հարցերի միջոցով պարզել, թե A և B բազմությունները հավասար են միմյանց:

1.15. *Թեորեմ: Բազմությունների հավասարությունը ստուգող լավագույն ալգորիթմ գոյություն ունի և նրա բարդությունը հավասար է* $\frac{n(n+1)}{2}$:

Ապացույց: Սկզբում նկարագրենք բազմությունների հավասարությունը ստուգող ալգորիթմ, որն ամենաշատը $\frac{n(n+1)}{2}$ հարցերի միջոցով լուծում է առաջադրված խնդիրը:

1.16. *Ալգորիթմ:*

(i) *Սահմանել A բազմության տարրերի $a_1, a_2, \dots, a_n$ հերթականությունը և որպես A բազմության հերթական տարր համարել a_1 տարրը:*

(ii) *Դիտարկել A բազմության հերթական տարրը և այն հաջորդաբար համեմատել B բազմության տարրերի հետ: Եթե B բազմությունում գոյություն ունի A բազմության հերթական տարրին հավասար տարր, ապա վերջինս հեռացնել B բազմությունից, իսկ հակառակ դեպքում՝ ավարտել ալգորիթմի աշխատանքը $A \neq B$ պատասխանով:*

(iii) *Եթե A բազմության հերթական տարրն a_n տարրն է, ապա ավարտել ալգորիթմի աշխատանքը $A = B$ պատասխանով, իսկ հակառակ դեպքում՝ որպես A բազմության հերթական տարր համարել հաջորդ տարրը և վերադառնալ* (ii) *քայլի կատարմանը:*

Առաջարկված ալգորիթմում հարցերի առավելագույն քանակ կօգտագործվի, երբ A բազմության ամեն մի տարրին հավասար տարրը B բազմությունից գտնվում է վերջին հնարավոր քայլում: Հետևաբար այդ ալգորիթմի բարդությունը կլինի

$$n + (n-1) + \dots + 2 + 1 = \frac{n(n+1)}{2}:$$

Թեորեմի ապացույցն ավարտելու համար բավական է բազմությունների հավասարությունը ստուգող ցանկացած ալգորիթմին համապատասխան ծառում նշել ճյուղ, որի երկարությունն առնվազն $\frac{n(n+1)}{2}$ է:

Այդ նպատակով կվարվենք հետևյալ կերպ. այս և հետագա որոշ խնդիրների համար կառաջարկենք որոշակի կանոններ, որոնք թույլ են տալիս խնդիրը լուծող ցանկացած ալգորիթմին համապատասխան ծառում ընտրել ճյուղ: **Նշենք, որ այդ կանոնների ընտրությունը կախված կլինի խնդիրներից, և ոչ թե խնդիրը լուծող ալգորիթմներից:** Ընդունված է ալգորիթմին համապատասխան ծառում ճյուղ ընտրելու այդ կանոններին անվանել **գուշակ**:

Դիտարկենք $(n \times n)$-չափի աղյուսակ, որի տողերին համապատասխանում են $a_1, a_2, \dots, a_n$ տարրերը, իսկ սյուներին՝ $b_1, b_2, \dots, b_n$ տարրերը: Աղյուսակի i-րդ տողի և j-րդ սյան համապատասխան վանդակում կգրենք $a_i = b_j$ հարցին՝ գուշակի թելադրած պատասխանը:

Աղյուսակի n վանդակներ կանվանենք **անկախ**, եթե նրանցից ցանկացած երկուսը գտնվում են տարբեր տողերում և տարբեր սյուներում: Աղյուսակի վանդակը կանվանենք **թույլատրելի**, եթե այդ վանդակում «ոչ» չի գրված (վանդակը կա՛մ դատարկ է, կա՛մ էլ նրանում գրված է «այո»): Ալգորիթմի աշխատանքի սկզբնական պահին աղյուսակի բոլոր վանդակները դատարկ են և, հետևաբար, թույլատրելի են:

Գուշակի՝ ճյուղ ընտրելու կանոնների նկարագիրը: Հերթական $a_i = b_j$ հարցին պատասխանել «ոչ» և աղյուսակի համապատասխան վանդակում գրել «ոչ», եթե դրանից հետո աղյուսակում կարելի է նշել n հատ անկախ, թույլատրելի վանդակ, հակառակ դեպքում՝ $a_i = b_j$ հարցին պատասխանել «այո» և աղյուսակի համապատասխան վանդակում գրել «այո»:

Պարզ է, որ այս ձևով սահմանված գուշակը (ալգորիթմին համապատասխան ծառում ճյուղ ընտրելու կանոնների բազմությունը) A և B բազմությունների հավասարությունը ստուգող կամայական ալգորիթմին համապատասխան ծառում միարժեքորեն ընտրում է որևէ ճյուղ: Ավելին, այդ ճյուղին համապատասխանում է «այո» (այսինքն՝ $A = B$) պատասխանը: Իրոք, քանի որ ընտրվածը ճյուղ է, ապա նրան համապատասխանում է ճիշտ մեկ պատասխան: Մյուս կողմից, ըստ գուշակի սահմանման, այն միշտ թողնում է n անկախ, թույլատրելի վանդակների ընտրության հնարավորություն, իսկ այդպիսի ամեն մի հնարավորությանը համապատասխանում է «այո» պատասխան, հետևաբար՝ ճյուղին համապատասխանում է «այո» պատասխան:

Այժմ դիտարկենք գուշակի կողմից լրացված աղյուսակը՝ մեր խնդիրը լուծող որևէ ալգորիթմի աշխատանքի ավարտի պահին: Աղյուսակի n անկախ վանդակներում գրված է «այո» (գուշակի վարվելակերպից հետևում է, որ «այո» պարունակող վանդակներն անկախ են, և քանի որ ալգորիթմն իր աշխատանքն ավարտել է $A = B$ պատասխանով, ապա յուրաքանչյուր տողում պետք է լինի «այո» պարունակող վանդակ), որոշ վանդակներում գրված է «ոչ», ընդ որում թույլատրելի վանդակներից այլ եղանակով հնարավոր չէ ընտրել n հատ անկախ վանդակ:

1.17. *Լեմմա: Եթե գուշակի կողմից լրացված աղյուսակի ցանկացած տող և ցանկացած սյուն պարունակում են առնվազն երկու թույլատրելի վանդակ, ապա n անկախ, թույլատրելի վանդակների ընտրության հնարավորությունը միակը չէ:*

Ապացույց: Ինչպես նշեցինք վերևում, գուշակի կողմից լրացված աղյուսակը միշտ պարունակում է n անկախ, թույլատրելի վանդակ: Դիտարկենք n անկախ, թույլատրելի վանդակների որևէ ընտրություն: Դիցուք առաջին տողից ընտրված է $(1, j_1)$ վանդակը: Դիտարկենք j_1 համարի սյունը: Այն պարունակում է մեկ ուրիշ՝ (i_1, j_1) թույլատրելի վանդակը: Բայց i_1 համարի տողը պարունակում է ընտրված անկախ վանդակներից մեկը՝ (i_1, j_2) վանդակը: Այդ դեպքում j_2 համարի սյունը պարունակում է (i_2, j_2) թույլատրելի վանդակը: Շարունակելով նման դատողությունները՝ կստանանք տողերի և սյուների հետևյալ հաջորդականությունը՝ $1, j_1, i_1, j_2, i_2, j_3, i_3, \dots$, որտեղ $(1, j_1), (i_1, j_2), (i_2, j_3), \dots$ վանդակները հանդիսանում են ընտրված անկախ, թույլատրելի վանդակները, իսկ $(i_1, j_1), (i_2, j_2), (i_3, j_3), \dots$ վանդակները՝ թույլատրելի վանդակներ են: Հասկանալի է, որ տողերի և սյուների նշված հաջորդականությունը չի կարող անվերջ շարունակվել, հետևաբար գոյություն կունենա տող կամ սյուն, որն այդ շարքում կկրկնվի: Ենթադրենք, որ առաջինը կրկնվում է i_k համարի տողը.

$$i_k, j_{k+1}, i_{k+1}, j_{k+2}, \dots, i_{k+l}, j_{k+l+1}, i_k:$$

Հաջորդականության կառուցման եղանակից բխում է, որ $(i_k, j_{k+1}), (i_{k+1}, j_{k+2}), \dots, (i_{k+l}, j_{k+l+1})$ վանդակները պատկանում են ընտրված անկախ, թույլատրելի վանդակների բազմությանը: Հեշտ է նկատել, որ եթե այդ վանդակները փոխարինենք $(i_{k+1}, j_{k+1}), (i_{k+2}, j_{k+2}), \dots, (i_k, j_{k+l+1})$ վանդակներով (որոնք գտնվում են նույն տողերում և սյուներում, ինչ նշվածները), ապա կստացվի n անկախ, թույլատրելի վանդակների մեկ ուրիշ, նոր ընտրություն: Լեմման ապացուցված է:

Վերադառնանք թեորեմի ապացույցին: Լեմմայից հետևում է, որ ալգորիթմի աշխատանքի ավարտի պահին գուշակի կողմից լրացված աղյուսակում գոյություն ունի տող կամ սյուն, որի $(n-1)$ վանդակներում գրված է «ոչ», իսկ մի վանդակում՝ «այո»: Ջնջենք «այո» վանդակն պարունակող տողը և սյունը: Նկատենք, որ հեռացված տողը և սյունը պարունակում են առնվազն n հարցի պատասխան: Ստացված աղյուսակում նույնպես $(n-1)$ անկախ, թույլատրելի վանդակներն ընտրվում են միարժեքորեն, ուստի նրանում գոյություն կունենա տող կամ սյուն, որի $(n-2)$ վանդակներում գրված է «ոչ», իսկ մեկ վանդակում՝ «այո»: Կրկին ջնջենք «այո» վանդակն պարունակող տողը և սյունը: Հեռացված տողը և սյունը պարունակում են առնվազն $(n-1)$ հարցի պատասխան: Շարունակելով տողի և սյան հեռացման պրոցեսը՝ կստանանք $(n \times n)$-չափի աղյուսակ, որի միակ վանդակում գրված է «այո»:

Այսպիսով, գուշակի կողմից լրացված աղյուսակը պարունակում է առնվազն

$$\geq n + (n-1) + \dots + 2 + 1 = \frac{n(n+1)}{2}$$

հարցի պատասխան: Հետևաբար գուշակի կողմից ընտրված ճյուղի երկարությունն առնվազն $\frac{n(n+1)}{2}$ է: Թեորեմն ապացուցված է ամբողջությամբ: ■

§ 1.5. ԽՄԲԱԿՑՈՒԹՅԱՆ ԿԱԶՄԻ ՈՐՈՇՈՒՄ (ՌՎ ՌՎ Է)

Դիտարկենք մարդկանց խմբակցություն, որի անդամներն ունեն $1, 2, \dots, n$ $(n \geq 3)$ համարները: Հայտնի է, որ խմբակցությունը բաղկացած է օրինավոր և անօրեն մարդկանցից, և օրինավոր անդամների քանակը $\geq \left\lceil\frac{n+1}{2}\right\rceil$, այսինքն՝ կեսից ավելին օրինավոր մարդիկ են: Բացի այդ, խմբակցության անդամներից յուրաքանչյուրը գիտի մյուսների ով լինելը:

Թույլատրվում է խմբակցության $\boldsymbol{i}$-րդ անդամին հարցնել. «*Օրինավո՞ր է արդյոք $\boldsymbol{j}$-րդ անդամը*», որտեղ $\mathbf{1} \leq \boldsymbol{i} \neq \boldsymbol{j} \leq \boldsymbol{n}$: Պայմանավորվենք վերջինս նշանակել $(\boldsymbol{i}, \boldsymbol{j})$ կարգավորված զույգով և անվանել $(\boldsymbol{i}, \boldsymbol{j})$ հարց: Եթե խմբակցության $\boldsymbol{i}$-րդ անդամն օրինավոր է, ապա նա $(\boldsymbol{i}, \boldsymbol{j})$ հարցին տալիս է միայն ճիշտ պատասխան, հակառակ դեպքում՝ նրա պատասխանն անկանխատեսելի է. կարող է լինել ճիշտ կամ սխալ: Մեր նպատակն է՝ նվազագույն թվով հարցերի միջոցով որոշել խմբակցության կազմը՝ պարզել, թե խմբակցության անդամներից յուրաքանչյուրն ով է:

$\mathbf{1.18}$. ***Թեորեմ:*** *Խմբակցության կազմն որոշող լավագույն ալգորիթմ գոյություն ունի և նրա $\boldsymbol{\alpha}(\boldsymbol{n})$ բարդությունը հավասար է $\boldsymbol{\alpha}(\boldsymbol{n}) \equiv \left\lfloor\frac{3(n-1)}{2}\right\rfloor$:*

Ապացույց: Վերևի՝ $\boldsymbol{\alpha}(\boldsymbol{n}) \leq \left\lfloor\frac{3(n-1)}{2}\right\rfloor$ գնահատականը կապացուցենք ինդուկցիայի եղանակով:

Դիցուք $\boldsymbol{n} = \mathbf{3}$ և այդ դեպքում որպես առաջին հարց համարենք $(\mathbf{1}, \mathbf{2})$ հարցը: Եթե այդ հարցի պատասխանը «այո» է (այսինքն՝ առաջին անդամը գտնում է, որ $\mathbf{2}$-րդ անդամն օրինավոր է), ապա $\mathbf{2}$-րդն իսկապես օրինավոր է. հակառակ դեպքում անօրեն կլիներ նաև $\mathbf{1}$-ինը, որը կհակասեր խնդրի պայմանին՝ խմբակցության անդամների կեսից ավելին օրինավոր են: Իսկ եթե $(\mathbf{1}, \mathbf{2})$ հարցի պատասխանը «ոչ» է, ապա $\mathbf{1}$-ին և $\mathbf{2}$-րդ անդամներից գոնե մեկն անօրեն է, ուստի $\mathbf{3}$-րդ անդամն օրինավոր է: Այսպիսով, $\boldsymbol{n} = \mathbf{3}$ դեպքում մեկ հարցից հետո կգտնենք օրինավոր անդամի, և նրան հարցնելով մյուսների մասին (ևս երկու հարց տալով)՝ կգտնենք, թե ով ով է: Հետևաբար $\boldsymbol{\alpha}(\mathbf{3}) \leq \mathbf{3}$:

Այժմ ենթադրենք, որ $\mathbf{3} \leq \boldsymbol{k} < \boldsymbol{n}$ դեպքում $\mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{k}$ անդամներով խմբակցության կազմը կարելի է որոշել ամենաշատը $\left\lfloor\frac{3(k-1)}{2}\right\rfloor$ հարցերի միջոցով, և դիտարկենք $\mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{n}$ անդամներով որևէ խմբակցություն: Ընտրենք նրա $\mathbf{1}$-ին անդամին և նրա մասին մյուսներից հարցնենք այնքան ժամանակ, քանի դեռ.

a) «ոչ» պատասխանների քանակը չի գերազանցում «այո» պատասխանների քանակին և
b) «այո» պատասխանների քանակը փոքր է $\left\lfloor\frac{n-1}{2}\right\rfloor$ թվից:

Եթե $\mathbf{1}$-ին անդամի վերաբերյալ հարցերի շարքը կանգ է առնում (a) կետի պայմանով, ապա գոյություն ունի այնպիսի ոչ բացասական $\boldsymbol{j}$ ամբողջ թիվ, որ $\boldsymbol{j} + \mathbf{1}$ հոգի $\mathbf{1}$-ին անդամի վերաբերյալ պատասխանել են «ոչ» (այսինքն գտնում են, որ $\mathbf{1}$-ին անդամն անօրեն է), իսկ $\boldsymbol{j}$ հոգի $\mathbf{1}$-ին անդամի վերաբերյալ պատասխանել են «այո» (գտնում են, որ նա օրինավոր է): Այդ դեպքում հարցմանը մասնակցած $\mathbf{2}\boldsymbol{j} + \mathbf{2}$ անդամներից առնվազն $\boldsymbol{j} + \mathbf{1}$ հոգի անօրեն են: Իրոք, եթե $\mathbf{1}$-ին անդամն օրինավոր է, ապա նրա մասին «ոչ» պատասխանողներն ($\boldsymbol{j} + \mathbf{1}$ հոգի) անօրեն են, իսկ եթե $\mathbf{1}$-ին անդամն անօրեն է, ապա անօրեն են նաև նրա մասին «այո» պատասխանողները ($\boldsymbol{j}$ հոգի): Դիտարկենք մնացած $\boldsymbol{n} - (\mathbf{2}\boldsymbol{j} + \mathbf{2})$ անդամների խմբակցությունը: Հասկանալի է, որ նրանցում օրինավորների քանակը մեծ է անօրենների քանակից: Հետևաբար այս դեպքում խնդիրը կարելի է լուծել հետևյալ երեք հաջորդական քայլերի միջոցով.

1) Համաձայն ինդուկցիայի ենթադրության, ամենաշատը $\left\lfloor\frac{3(n-(2j+2)-1)}{2}\right\rfloor$ հարցերի միջոցով կորոշենք $\boldsymbol{n} - (\mathbf{2}\boldsymbol{j} + \mathbf{2})$ անդամներից բաղկացած խմբակցության կազմը.
2) Կորոշենք $\mathbf{1}$-ին անդամի ով լինելը՝ նրա մասին հարցնելով որևէ օրինավոր անդամի.
3) Եվ ամենաշատը $\boldsymbol{j} + \mathbf{1}$ հարցերի միջոցով կորոշենք մնացած $\mathbf{2}\boldsymbol{j} + \mathbf{1}$ անդամների ով լինելը. եթե $\mathbf{1}$-ին անդամն օրինավոր է, ապա «ոչ» պատասխանողներն անօրեն են, իսկ եթե $\mathbf{1}$-ին անդամն անօրեն է, ապա «այո» պատասխանողներն նույնպես անօրեն են:

Այսպիսով, այս դեպքում խմբակցության կազմն որոշելու համար անհրաժեշտ հարցերի քանակը չի գերազանցում

$$\mathbf{2}\boldsymbol{j} + \mathbf{1} + \left\lfloor\frac{\mathbf{3}(\boldsymbol{n} - (\mathbf{2}\boldsymbol{j} + \mathbf{2}) - \mathbf{1})}{\mathbf{2}}\right\rfloor + \mathbf{1} + \boldsymbol{j} + \mathbf{1} = \left\lfloor\frac{\mathbf{3}(\boldsymbol{n} - \mathbf{1})}{\mathbf{2}}\right\rfloor$$

մեծությունը:

Այժմ դիտարկենք այն դեպքը, երբ $\mathbf{1}$-ին անդամի վերաբերյալ հարցերի շարքը կանգ է առնում (b) կետի պայմանով: Վերջինս նշանակում է, որ ճիշտ $\left\lfloor\frac{n-1}{2}\right\rfloor$ անգամ ստացվել է «այո» պատասխան, իսկ $\boldsymbol{j}$ $\left(\mathbf{0} \leq \boldsymbol{j} \leq \left\lfloor\frac{n-1}{2}\right\rfloor\right)$ անգամ՝ «ոչ» պատասխան: Հետևաբար $\mathbf{1}$-ին անդամն օրինավոր է: Իսկապես, եթե նա անօրեն է, ապա այդպիսին են նաև $\left\lfloor\frac{n-1}{2}\right\rfloor$ հոգի՝ «այո» պատասխանողները, ուստի անօրենների քանակը կլինի $\geq \left\lfloor\frac{n-1}{2}\right\rfloor + \mathbf{1} = \left\lfloor\frac{n+1}{2}\right\rfloor$, որը հակասում է խնդրի պայմանին: Այստեղից հետևում է, որ «ոչ» պատասխանողները ($\boldsymbol{j}$ հոգի) անօրեն են:

Առաջին անդամին տալով $\boldsymbol{n-1-j}$ հարց մնացած անդամների մասին՝ կորոշենք խմբակցության կազմը: Այս դեպքում հարցերի քանակը կլինի

$$j+\left\lfloor\frac{n-1}{2}\right\rfloor+n-1-j=\left\lfloor\frac{3(n-1)}{2}\right\rfloor:$$

Հետևաբար $\alpha(n)\le\left\lfloor\frac{3(n-1)}{2}\right\rfloor$, երբ $n\ge 3$:

Արդ ցույց տանք, որ $\alpha(n)\ge\left\lfloor\frac{3(n-1)}{2}\right\rfloor$, երբ $n\ge 3$: Դրա համար բավական է ապացուցել, որ խնդիրը լուծող ցանկացած ալգորիթմին համապատասխան ծառում կարելի է նշել առնվազն $\left\lfloor\frac{3(n-1)}{2}\right\rfloor$ երկարություն ունեցող ճյուղ:

Խնդիրը լուծող ցանկացած ալգորիթմին համապատասխան ծառում սահմանենք ճյուղ ընտրելու եղանակը՝ գուշակը:

Ալգորիթմի առաջին $k=\left\lfloor\frac{n-1}{2}\right\rfloor-1$ հարցերին՝ $h_1=(i_1,j_1)$, $h_2=(i_2,j_2)$, ... , $h_k=(i_k,j_k)$, պատասխանել «ոչ»: Մնացած հարցերի պատասխաններն ընտրելուց առաջ դիտարկենք $G=(V,X)$ գրաֆը, որտեղ

$$V=\{i_1,j_1\}\cup\{i_2,j_2\}\cup\dots\cup\{i_k,j_k\}\quad \text{և}\quad X=\{\{i_1,j_1\},\{i_2,j_2\},\dots,\{i_k,j_k\}\}:$$

Դիցուք G գրաֆի կապակցվածության բաղադրիչներ հանդիսանում են $G_l=(V_l,X_l)$, $l=1,2,\dots,r$, ենթագրաֆները և $W=\{1,2,\dots,n\}\setminus V$: Հաջորդ $\{h_{k+1},h_{k+2},\dots\}$ հարցերի պատասխաններն ընտրենք հետևյալ կերպ. երբ $s>k$, ապա $h_s=(i_s,j_s)$ հարցին պատասխանել

- «այո» , եթե $j_s\in W$;
- «ոչ» , եթե $j_s\in V_l(1\le l\le r)$ և գոյություն ունի V_l բազմության տարր, որի մասին $\{h_{k+1},h_{k+2},\dots\}$ հարցերի ժամանակ հարցում չի կատարվել;
- «այո» , եթե $j_s\in V_l(1\le l\le r)$ և գոյություն չունի V_l բազմության տարր, որի մասին $\{h_{k+1},h_{k+2},\dots\}$ հարցերի ժամանակ հարցում չի կատարվել:

Սրանով գուշակի նկարագիրն ավարտված է, ըստ որի խնդիրը լուծող յուրաքանչյուր ալգորիթմին համապատասխան ծառում ընտրվում է որոշակի ճյուղ:

Հաջորդիվ, խմբակցության այն անդամների բազմությունը, որոնց մասին $\{h_{k+1},h_{k+2},\dots\}$ հարցերի ժամանակ «ոչ» պատասխան չի տրվել (կա՛մ հարցում չի կատարվել, կա՛մ էլ տրվել է միայն «այո» պատասխան) նշանակենք L տառով: Պարզ է, որ բոլոր $l=1,2,\dots,r$ համար $|L\cap V_l|\ge 1$:

Սահմանենք L^* բազմությունը հետևյալ կանոնների միջոցով.

1) $W\subseteq L^*$;
2) Եթե $j\in V_l$ և j անդամի մասին $\{h_{k+1},h_{k+2},\dots\}$ հարցերի ժամանակ տրվել է միայն «այո» պատասխան, ապա $j\in L^*$;
3) Եթե V_l բազմության յուրաքանչյուր տարրի մասին $\{h_{k+1},h_{k+2},\dots\}$ հարցերի ժամանակ չի տրվել «այո» պատասխան (կա՛մ հարցում չի կատարվել, կա՛մ էլ տրվել է միայն «ոչ» պատասխան), ապա $L\cap V_l$ բազմության ամենափոքր j տարրն ավելացնել L^* բազմությանը, այսինքն՝ $j\in L^*$;
4) Այլ կանոններ չկան և L^* բազմությունն որոշվում է վերը նկարագրված $(1)-(3)$ կանոնների միջոցով:

Նկատենք, որ L^* բազմության սահմանումից բխում է, որ բոլոր $l=1,2,\dots,r$ համար $|L^*\cap V_l|=1$, և

$$|L^*\cap V|=|L^*\cap V_1|+|L^*\cap V_2|+\dots+|L^*\cap V_r|=\underbrace{1+1+\dots+1}_{r}=r:$$

Մյուս կողմից, քանի որ $G_l=(V_l,X_l)$ գրաֆները կապակցված են, ապա

$$|X_l|\ge|V_l|-1$$

բոլոր $l=1,2,\dots,r$ համար: Հետևաբար

$$k=|X|=|X_1|+|X_2|+\dots+|X_r|\ge|V_1|+|V_2|+\dots+|V_r|-r=|V|-r$$

կամ, որ նույնն է, $|V|\le k+r$: Սրանից հետևում է, որ

$$|V\setminus L^*|=|V|-|V\cap L^*|\le k+r-r=k:$$

Ուստի ունենք, որ

$$|L^*|=|L^*\cap W|+|L^*\cap V|=n-|V|+|V|-|V\setminus L^*|=n-|V\setminus L^*|\ge n-k=$$

$$=n-\left\lfloor\frac{n-1}{2}\right\rfloor+1:$$

Ցույց տանք, որ գուշակի ընտրած ճյուղի երկարությունն առնվազն $\left\lfloor\frac{3(n-1)}{2}\right\rfloor$ է: Ենթադրենք հակառակը: Քանի որ

$$\left\lfloor\frac{3m}{2}\right\rfloor-\left\lfloor\frac{m}{2}\right\rfloor=m\,,$$

ապա

$$\left\lfloor\frac{3(n-1)}{2}\right\rfloor = k + \left\lfloor\frac{3(n-1)}{2}\right\rfloor - \left\lfloor\frac{n-1}{2}\right\rfloor + 1 = k + n$$

հավասարությունից հետևում է, որ $\{h_{k+1}, h_{k+2}, \dots\}$ հարցերի քանակը չի գերազանցում $n-1$ թիվը: Դա նշանակում է, որ գոյություն ունի խմբակցության a անդամ, որի մասին $\{h_{k+1}, h_{k+2}, \dots\}$ հարցերի ժամանակ հարցում չի կատարվել: Պարզ է, որ $a \in L$: Մյուս կողմից, առանց ընդհանրությունը խախտելու, կարող ենք ենթադրել, որ $a \in L^*$: Իսկապես, դիցուք $a \notin L^*$: Ուրեմն $a \notin W$ և, հետևաբար, $a \in V_l$ որևէ $1 \le l \le r$ համար, ընդ որում, ըստ գուշակի սահմանման, $G_l = (V_l, X_l)$ բաղադրիչի ոչ մի տարրի մասին չի տրվել «այո» պատասխան: Վերջինս նշանակում է, որ L^* բազմության՝ $G_l = (V_l, X_l)$ բաղադրիչին պատկանող տարրն ընտրվել է L^* բազմության սահմանման (3) կետով: Կազմակերպելով խմբակցության անդամների վերահամարակալում՝ կարող ենք հասնել նրան, որ $L \cap V_l$ բազմության ամենափոքր տարրը հանդիսանա a անդամը, ուստի $a \in L^*$:

Նկարագրենք երկու իրավիճակ.

Իրավիճակ Ա: L^* բազմությանը պատկանող խմբակցության անդամներն օրինավոր են, իսկ մնացածները՝ անօրեն: Պարզ է, որ անօրենների քանակը $\le \left\lfloor\frac{n-1}{2}\right\rfloor - 1$:

Իրավիճակ Բ: L^* բազմությանը պատկանող խմբակցության անդամներն, բացի a անդամից, օրինավոր են, իսկ մնացածները՝ անօրեն: Այս դեպքում անօրենների քանակը $\le \left\lfloor\frac{n-1}{2}\right\rfloor$:

Դժվար չէ նկատել, որ *Ա* և *Բ* իրավիճակները բավարարում են գուշակի կողմից ընտրված ճյուղի բոլոր հարցերին և, հետևաբար, խնդիրը լուծող ալգորիթմը չի տարբերում այդ իրավիճակները: Ստացված հակասությունն ապացուցում է թեորեմը:

Դիտողություն: Խմբակցության անդամների կեսից ավելին օրինավոր լինելու պայմանն էական է, քանի որ հակառակ դեպքում անօրեններն իրենց պատասխանները կարող էին կազմակերպել այնպես, որ անօրենների ցանկացած խումբ ընկալվեր որպես օրինավորների խումբ:

Ավելի հաճախ հանդիպում է քննարկվող խնդրի այն տարբերակը, երբ խմբակցության անդամներից յուրաքանչյուրը կա՛մ ճշտախոս է, կա՛մ էլ ստախոս (միշտ ստում է. ցանկացած հարցի տալիս է սխալ պատասխան): Այդպիսի խմբակցության կազմն որոշելու խնդիրը, երբ ճշտախոսները կեսից ավելի են, շատ հեշտ է: Իսկապես, եթե ընտրենք որևէ մեկին և նրան հարցնենք մնացածների

մասին, ապա ստացված $n-1$ պատասխաններով կորոշենք, թե ով ով է: Հեշտ է նկատել, որ այս դեպքում խմբակցության կազմ որոշող հարցերի քանակը չի կարող փոքր լինել $n-1$ թվից:

Գլուխ 2

ՄՐՑԱՇԱՐԱՅԻՆ ԽՆԴԻՐՆԵՐ

§ 2.1. ԽՆԴԻՐՆԵՐԻ ՄԵԿՆԱԲԱՆՈՒՄԸ: ՄՐՑԱՇԱՐԻ ՀԱՂԹՈՂԻ (ՊԱՐՏՎՈՂԻ) ՈՐՈՇՈՒՄ

2.1. ***Սահմանում:*** *$\vec{G} = (V, E)$ կողմնորոշված գրաֆը (օրգրաֆը) կոչվում է* ***մրցաշար,*** *եթե ցանկացած $u, v \in V$ ($u \neq v$) գագաթների համար տեղի ունի հետևյալ պնդումը.*

$$(u, v) \in E \Leftrightarrow (v, u) \notin E:$$

Այս անվանումը կապված է սպորտային մրցումների արդյունքներն օրգրաֆների միջոցով ներկայացման հետ: Դիտարկենք մեկ փուլով անցկացվող ֆուտբոլի կամ որևէ այլ սպորտային մրցաշար, որում մասնակցում են $1, 2, \ldots, n$ համարներն ունեցող թիմերը (խաղացողները, մասնակիցները և այլն): Սահմանենք $\vec{G} = (V, E)$ կողմնորոշված գրաֆը, որում $V = \{1, 2, \ldots, n\}$ և $(i, j) \in E$, եթե $i \neq j$ և մրցաշարի ընթացքում i-րդ թիմը հաղթել է j-րդ թիմին: Պարզ է, որ այդ օրգրաֆը բավարարում է վերը նշված պայմաններին:

2.2. ***Սահմանում:*** *$\vec{G} = (V, E)$ մրցաշարը կոչվում է* ***տրանզիտիվ,*** *եթե ցանկացած $u, v, w \in V$ գագաթների համար տեղի ունի հետևյալ պայմանը.*

$$\big((u, v) \in E, (v, w) \in E\big) \Rightarrow \big((u, w) \in E\big):$$

Հետագայում կդիտարկենք միայն տրանզիտիվ մրցաշարեր, և մրցաշար ասելով՝ կհասկանանք տրանզիտիվ մրցաշար: Մրցաշարային խնդիրներն ունեն տարբեր մեկնաբանություններ: Նկարագրենք նրանցից երկուսը:

a) Դիտարկենք մրցաշար, որին մասնակցում են $a_1, a_2, \ldots, a_n$ խաղացողները: Ենթադրենք, որ նրանք տարբեր ուժի խաղացողներ են, բայց մենք տեղեկություն չունենք նրանց ուժի մասին: Թույլատրվում է ցանկացած երկու խաղացողների միջև կազմակերպել խաղ, որի արդյունքում կորոշվեն ուժեղ և թույլ խաղացողները: Ենթադրվում է, որ ուժեղ խաղացողը միշտ հաղթում է թույլ խաղացողին: Պահանջվում է նվազագույն թվով խաղերի կազմակերպման միջոցով որոշել մրցաշարի հաղթողին (պարտվողին, հաղթողին և պարտվողին, առաջին երկու տեղերը գրավողներին, առաջին երեք տեղերը գրավողներին և այլն):

b) Դիցուք ունենք $a_1, a_2, \ldots, a_n$ միանման կշռաքարերը, որոնց քաշերը զույգ առ զույգ տարբեր են և, կշռաքարերին նայելով, հնարավոր չէ համեմատել այդ քաշերը: Տրված է նաև նժարավոր կշեռք, որի միջոցով կարող ենք համեմատել ընտրված երկու կշռաքարերի քաշերը: Պահանջվում է նվազագույն թվով կշռումների միջոցով որոշել ամենածանր (ամենաթեթև, ամենածանր և ամենաթեթև, երկու ամենածանր, երեք ամենածանր և այլն) կշռաքար(եր)ը:

Դժվար չէ նկատել, որ իրականում այս երկու մեկնաբանությունները հանդիսանում են միևնույն մրցաշարային խնդիրը:

Այժմ քննարկենք մրցաշարի հաղթողի որոշման խնդիրը (a) մեկնաբանությամբ: Դիտարկենք խնդիրը լուծող երկու ալգորիթմ:

2.3. *Ալգորիթմ (սովորական մրցակարգ):*

(i) Որպես հաղթողի հավակնորդ ընտրել a_1 խաղացողին, իսկ որպես հերթական մասնակից՝ a_2 խաղացողին:

(ii) Հաղթողի հավակնորդից և հերթական մասնակցից որոշել ուժեղին և նրան համարել հաղթողի հավակնորդ:

(iii) Եթե հերթական մասնակիցը a_n խաղացողն է, ապա ավարտել ալգորիթմի աշխատանքը «հավակնորդը մրցաշարի հաղթողն է» պատասխանով, հակառակ դեպքում՝ հաջորդ խաղացողին համարել հերթական մասնակից և վերադառնալ (ii) քայլի կատարմանը:

Պարզ է, որ նշված ալգորիթմը $n - 1$ խաղերից հետո կորոշի մրցաշարի հաղթողին:

2.4. *Ալգորիթմ (զավաթային մրցակարգ):*

(i) Մրցաշարին մասնակցող յուրաքանչյուր խաղացողի համարել հաղթողի հավակնորդ:

(ii) Հաղթողի հավակնորդներին տրոհել զույգերի (եթե նրանց քանակը կենտ է, ապա առանձնացնել որևէ մեկին, իսկ մնացածներին տրոհել զույգերի): Յուրաքանչյուր զույգում որոշել ուժեղ խաղացողին և նրան համարել հաղթողի հավակնորդ:

(iii) Եթե հաղթողի հավակնորդների քանակը մեկ է, ապա ավարտել ալգորիթմի աշխատանքը «հաղթողի միակ հավակնորդը մրցաշարի հաղթողն է» պատասխանով, հակառակ դեպքում՝ վերադառնալ (ii) քայլի կատարմանը:

Մաթեմատիկական ինդուկցիայի եղանակով ցույց տանք, որ զավաթային մրցակարգում ևս մրցաշարի հաղթողին որոշելու համար պահանջվում է կազմակերպել $n - 1$ խաղ, եթե մասնակիցների ընդհանուր քանակը n է: Երբ $n =$

$\mathbf{1}$, ապա պնդումն ակնհայտ է: Ենթադրենք այն ճիշտ է $< \boldsymbol{n}$ դեպքում և ապացուցենք $\boldsymbol{n}$ համար: Համաձայն ալգորիթմի, առաջին փուլում կանցկացվի $\lfloor \boldsymbol{n}/\mathbf{2} \rfloor$, իսկ երկրորդ փուլում կլինեն $\lceil \boldsymbol{n}/\mathbf{2} \rceil$ հավակնորդներ: Հետևաբար, ըստ ինդուկցիայի ենթադրության, $\lceil \boldsymbol{n}/\mathbf{2} \rceil$ հավակնորդներից հաղթողին գավաթային մրցակարգը կորոշի $\lceil \boldsymbol{n}/\mathbf{2} \rceil - \mathbf{1}$ խաղերի արդյունքում: Ուստի ընդհանուր խաղերի քանակը կլինի

$$\left\lfloor \frac{\boldsymbol{n}}{\mathbf{2}} \right\rfloor + \left\lceil \frac{\boldsymbol{n}}{\mathbf{2}} \right\rceil - \mathbf{1} = \boldsymbol{n} - \mathbf{1}:$$

$\mathbf{2.5}$. *Լեմմա: Եթե մրցաշարի մասնակիցների քանակը* $\boldsymbol{n}$ *է, ապա գավաթային մրցակարգում յուրաքանչյուր խաղացող մասնակցում է ամենաշատը* $\lceil \mathbf{log_2}\, \boldsymbol{n} \rceil$ խաղի:

Ապացույց: Իսկապես, քանի որ $\boldsymbol{n} \le \mathbf{2}^{\lceil \mathbf{log_2}\, \boldsymbol{n} \rceil}$, ապա գավաթային մրցակարգում փուլերի քանակը չի գերազանցում $\lceil \mathbf{log_2}\, \boldsymbol{n} \rceil$ թիվը, իսկ ամեն մի փուլում յուրաքանչյուր մասնակից խաղում է ամենաշատը մեկ անգամ: ■

Մրցաշարի $\boldsymbol{n}$ մասնակիցներից մրցաշարի հաղթողին որոշող լավագույն ալգորիթմի բարդությունը նշանակենք $\boldsymbol{W_1(n)}$ գրությամբ:

$\mathbf{2.6}$. *Թեորեմ:* $\boldsymbol{W_1(n) = n - 1}$:

Ապացույց: Վերևում նկարագրված ալգորիթմները ցույց են տալիս, որ $\boldsymbol{W_1(n) \le n - 1}$: Մյուս կողմից, եթե մրցաշարի ավարտին հաղթողն որոշված է, ապա մնացած $\boldsymbol{n} - \mathbf{1}$ խաղացողներից յուրաքանչյուրը պետք է ունենա գոնե մեկ պարտություն, որը բնութագրվում է մեկ խաղով և, հետևաբար, մրցաշարի հաղթողին որոշելու համար անհրաժեշտ խաղերի քանակը $\ge (\boldsymbol{n} - \mathbf{1})$: Այսպիսով $\boldsymbol{W_1(n) = n - 1}$: ■

Հեշտությամբ կարելի է համոզվել, որ $\boldsymbol{n}$ մասնակիցներ ունեցող մրցաշարի ամենաթույլ խաղացողին (պարտվողին) կարելի է որոշել ճիշտ $\boldsymbol{n} - \mathbf{1}$ խաղերի արդյունքում:

§ 2.2. ՄՐՑԱՇԱՐԻ ՀԱՂԹՈՂԻ ԵՎ ՊԱՐՏՎՈՂԻ ՈՐՈՇՈՒՄ

Մրցաշարային խնդիրներում $\boldsymbol{n}$ խաղացողներից մրցաշարի հաղթողին և պարտվողին որոշող լավագույն ալգորիթմի բարդությունը նշանակենք $\boldsymbol{U(n)}$:

$\mathbf{2.7}$. *Թեորեմ:* $\boldsymbol{U(n)} = \lceil \mathbf{3}\boldsymbol{n}/\mathbf{2} \rceil - \mathbf{2}$:

Ապացույց: Զույգ թվով մասնակիցների՝ $\boldsymbol{n} = \mathbf{2}\boldsymbol{k}$, դեպքում մրցաշարի հաղթողին և պարտվողին կարելի է գտնել հետևյալ եղանակով.

(i) Մրցաշարի մասնակիցներին տրոհում ենք $\boldsymbol{k}$ զույգերի և յուրաքանչյուր զույգում որոշում հաղթողին և պարտվողին՝ կազմակերպելով $\boldsymbol{k}$ խաղ:
(ii) Զույգերի $\boldsymbol{k}$ հաղթողներից՝ $\boldsymbol{k} - \mathbf{1}$ խաղերի միջոցով որոշում ենք ամենաուժեղին:
(iii) Զույգերի $\boldsymbol{k}$ պարտվողներից՝ $\boldsymbol{k} - \mathbf{1}$ խաղերի միջոցով որոշում ենք ամենաթույլին:

Այս դեպքում խաղերի ընդհանուր քանակը հավասար է $\mathbf{3}\boldsymbol{k} - \mathbf{2} = \lceil \mathbf{3}\boldsymbol{n}/\mathbf{2} \rceil - \mathbf{2}$:

Նույն եղանակը կարելի է կիրառել $\boldsymbol{n} = \mathbf{2}\boldsymbol{k} + \mathbf{1}$ դեպքում՝ նախապես առանձնացնելով մեկ խաղացող և նրան մասնակից դարձնելով (ii) և (iii) կետերի խաղերին: Այս դեպքում ևս հաղթողին և պարտվողին կարելի է գտնել $\mathbf{3}\boldsymbol{k} = \lceil \mathbf{3}\boldsymbol{n}/\mathbf{2} \rceil - \mathbf{2}$ խաղերի միջոցով: Հետևաբար $\boldsymbol{U(n)} \le \lceil \mathbf{3}\boldsymbol{n}/\mathbf{2} \rceil - \mathbf{2}$:

Ստորին՝ $\boldsymbol{U(n)} \ge \lceil \mathbf{3}\boldsymbol{n}/\mathbf{2} \rceil - \mathbf{2}$ գնահատականի ապացույցի համար օգտվենք նախորդ գլխում ներմուծված գուշակի գաղափարից (կանոնների բազմություն), որը թույլ է տալիս խնդիրը լուծող ծանկացած ալգորիթմին համապատասխան ծառում ընտրել որոշակի ճյուղ: Գուշակի նպատակն ալգորիթմի աշխատանքի երկարաձգումն է:

Դիտարկենք մրցաշարի հաղթողին և պարտվողին որոշող որևէ ալգորիթմ: Այդ ալգորիթմի աշխատանքի ընթացքում մրցաշարի մասնակիցներին կվերագրենք $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{C}}$ և $\boldsymbol{\mathcal{D}}$ հատկություններից որևէ մեկը.

$\boldsymbol{\mathcal{A}}$ – այն խաղացողներին, որոնք ոչ մի խաղի դեռ չեն մասնակցել,

$\boldsymbol{\mathcal{B}}$ – այն խաղացողներին, որոնք ունեն միայն հաղթանակներ,

$\boldsymbol{\mathcal{C}}$ – այն խաղացողներին, որոնք ունեն միայն պարտություններ,

$\boldsymbol{\mathcal{D}}$ – այն խաղացողներին, որոնք ունեն և՛ հաղթանակ, և՛ պարտություն:

Գուշակի վարվելակերպը սահմանենք ստորև բերված աղյուսակի առաջին երկու սյուներին համապատասխան:

Եթե հանդիպում են նշված հատկությամբ մասնակիցներ	Գուշակի ընտրած արդյունքը	Գուշակի ընտրության արդյունքում $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{C}}$ և $\boldsymbol{\mathcal{D}}$ հատկություններով օժտված մասնակիցների քանակների փոփոխությունը			
		$\boldsymbol{\mathcal{A}}$	$\boldsymbol{\mathcal{B}}$	$\boldsymbol{\mathcal{C}}$	$\boldsymbol{\mathcal{D}}$
$\boldsymbol{\mathcal{A} \vdots \mathcal{A}}$	կամայական	-2	$+1$	$+1$	0
$\boldsymbol{\mathcal{A} \vdots \mathcal{B}}$	$\boldsymbol{\mathcal{A} < \mathcal{B}}$	-1	0	$+1$	0
$\boldsymbol{\mathcal{A} \vdots \mathcal{C}}$	$\boldsymbol{\mathcal{A} > \mathcal{C}}$	-1	$+1$	0	0
$\boldsymbol{\mathcal{A} \vdots \mathcal{D}}$	$\boldsymbol{\mathcal{A} > \mathcal{D}}$	-1	$+1$	0	0
$\boldsymbol{\mathcal{B} \vdots \mathcal{B}}$	կամայական	0	-1	0	$+1$
$\boldsymbol{\mathcal{B} \vdots \mathcal{C}}$	$\boldsymbol{\mathcal{B} > \mathcal{C}}$	0	0	0	0
$\boldsymbol{\mathcal{B} \vdots \mathcal{D}}$	$\boldsymbol{\mathcal{B} > \mathcal{D}}$	0	0	0	0
$\boldsymbol{\mathcal{C} \vdots \mathcal{C}}$	կամայական	0	0	-1	$+1$
$\boldsymbol{\mathcal{C} \vdots \mathcal{D}}$	$\boldsymbol{\mathcal{C} < \mathcal{D}}$	0	0	0	0
$\boldsymbol{\mathcal{D} \vdots \mathcal{D}}$	կամայական	0	0	0	0

Պարզ է, որ գուշակը, մրցաշարի հաղթողին և պարտվողին որոշող ցանկացած ալգորիթմում, ընտրում է որոշակի ճյուղ: Այդ ճյուղի ընտրությունից հետո ալգորիթմում տեղի ունեցած $\boldsymbol{\mathcal{A} \vdots \mathcal{A}}$, $\boldsymbol{\mathcal{A} \vdots \mathcal{B}}$, $\boldsymbol{\mathcal{A} \vdots \mathcal{C}}$, $\boldsymbol{\mathcal{A} \vdots \mathcal{D}}$, $\boldsymbol{\mathcal{B} \vdots \mathcal{B}}$ և $\boldsymbol{\mathcal{C} \vdots \mathcal{C}}$ տիպի խաղերի քանակները համապատասխանաբար նշանակենք $\boldsymbol{x_1, x_2, x_3, x_4, x_5}$ և $\boldsymbol{x_6}$, իսկ մնացած խաղերի քանակը՝ $\boldsymbol{y}$: Այդ դեպքում ալգորիթմում՝ գուշակի կողմից ընտրված ճյուղում խաղերի քանակը հավասար է $\boldsymbol{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + y}$:

Բերված աղյուսակի վերջին չորս սյուներում նշված է $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{C}}$ և $\boldsymbol{\mathcal{D}}$ հատկություններով օժտված մասնակիցների փոփոխությունները՝ տեղի ունեցած խաղերից հետո:

Ալգորիթմի աշխատանքի սկզբում $\boldsymbol{\mathcal{A}}$ հատկությամբ օժտված մասնակիցների քանակը $\boldsymbol{n}$ էր, իսկ ավարտի պահին՝ $\boldsymbol{0}$, ուստի $\boldsymbol{n - (2x_1 + x_2 + x_3 + x_4) = 0}$ կամ $\boldsymbol{(2x_1 + x_2 + x_3 + x_4) = n}$: Մյուս կողմից, ալգորիթմի աշխատանքի սկզբում $\boldsymbol{\mathcal{D}}$ հատկությամբ օժտված մասնակիցների քանակը $\boldsymbol{0}$ էր,

իսկ ավարտի պահին՝ $\boldsymbol{n-2}$: Այնպես որ $\boldsymbol{0 + x_5 + x_6 = n - 2}$ կամ $\boldsymbol{x_5 + x_6 = n - 2}$: Հետևաբար

$$\boldsymbol{2(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + y) \geq (2x_1 + x_2 + x_3 + x_4) + 2(x_5 + x_6) = 3n - 4}$$

և

$$\boldsymbol{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + y \geq \lceil 3n/2 \rceil - 2}:$$

Այսպիսով, խնդիրը լուծող ցանկացած ալգորիթմում նշեցինք դեպք, որում տեղի ունեցած խաղերի քանակն առնվազն $\boldsymbol{\lceil 3n/2 \rceil - 2}$ է: Ուստի $\boldsymbol{U(n) = \lceil 3n/2 \rceil - 2}$: ■

§ 2.3. ՄՐՑԱՇԱՐԻ ԱՌԱՋԻՆ ԵՎ ԵՐԿՐՈՐԴ ՏԵՂԵՐԸ ԳՐԱՎՈՂՆԵՐԻ ՈՐՈՇՈՒՄ

Դիտարկենք մրցաշար, որին մասնակցում են $\boldsymbol{n}$ խաղացողներ ($\boldsymbol{n \geq 2}$) և քննարկենք հետևյալ խնդիրը. **Անհրաժեշտ է, նվազագույն թվով խաղեր կազմակերպելով, գտնել մրցաշարի երկու հաղթողներին՝ առաջին և երկրորդ տեղերը գրավողներին:**

Խնդիրը լուծող լավագույն ալգորիթմի բարդությունը նշանակենք $\boldsymbol{W_2(n)}$:

2.8. *Թեորեմ:* $\boldsymbol{W_2(n) = n - 2 + \lceil \log_2 n \rceil}$:

Ապացույց: Մրցաշարում երկրորդ տեղը գրավողը պետք է գտնվի այն մասնակիցների մեջ, որոնք պարտվել են միայն մրցաշարի հաղթողին: Հետևաբար, մրցաշարի առաջին և երկրորդ տեղերը գրավողներին կարելի է գտնել հետևյալ եղանակով.

(i) Գավաթային մրցակարգով, կազմակերպելով $\boldsymbol{(n-1)}$ խաղ, գտնում ենք մրցաշարի հաղթողին.

(ii) Մրցաշարի հաղթողին պարտված մասնակիցներից որոշում ենք ամենաուժեղին (այն կլինի մրցաշարի երկրորդ տեղը գրավողը):

Քանի որ գավաթային մրցակարգում հաղթողը մասնակցում է առավելագույնը $\boldsymbol{\lceil \log_2 n \rceil}$ խաղի, ուստի $\boldsymbol{W_2(n) \leq n - 1 + \lceil \log_2 n \rceil - 1 = n - 2 + \lceil \log_2 n \rceil}$:

Այժմ ցույց տանք, որ $\boldsymbol{W_2(n) \geq n - 2 + \lceil \log_2 n \rceil}$: Դիտարկենք մրցաշարի երկու հաղթողներին որոշող որևէ ալգորիթմ: Հասկանալի է, որ առանց մրցաշարի հաղթողին որոշելու՝ հնարավոր չէ որոշել երկրորդ տեղը զբաղեցնողին: Վերջինս նշանակում է, որ բացի մրցաշարի հաղթողից, մնացած բոլոր մասնակիցներն ունեն առնվազն մեկ պարտություն: Մրցաշարի մասնակիցներին համապատասխանեցնենք $x_1, x_2, \cdots, x_n$ նշանները և, առանց ընդհանրությունը խախտելու, ենթադրենք, որ x_n մասնակիցը մրցաշարի հաղթողն է: Կազմենք հետևյալ ցուցակը.

x_1 մասնակիցը մրցաշարի ընթացքում ունի k_1 ($k_1 \ge 1$) պարտություն,

x_2 մասնակիցը մրցաշարի ընթացքում ունի k_2 ($k_2 \ge 1$) պարտություն,

$\vdots$

x_{n-1} մասնակիցը մրցաշարի ընթացքում ունի k_{n-1} ($k_{n-1} \ge 1$) պարտություն:

Քանի որ յուրաքանչյուր պարտություն համապատասխանում է մրցաշարի ճիշտ մեկ խաղի, ապա $K = k_1 + k_2 + \cdots + k_{n-1}$ թիվը հանդիսանում է այդ մրցաշարի բոլոր խաղերի քանակը: Կատարենք հետևյալ նշանակումը.

$$a_j = |\{x_i | k_i \ge j;\ i = 1, 2, \dots, n-1\}|,\ j = 1, 2, 3, \dots n-1 :$$

Հեշտ է համոզվել, որ a_j հանդիսանում է ալգորիթմի աշխատանքի ավարտի պահին առնվազն j պարտություն ունեցող մասնակիցների քանակը, $j = 1, 2, \dots, n-1$: Այդ դեպքում $K = a_1 + a_2 + \cdots + a_{n-1}$: Պարզ է, որ $a_1 = n-1$: Ենթադրենք, որ մրցաշարի հաղթողը խաղացել է p խաղ: Ակնհայտ է, որ նրան պարտվողներից որևէ մեկը պետք է զբաղեցնի երկրորդ տեղը և հետևաբար $a_2 \ge p-1$: Թեորեմն ապացուցելու համար բավական է կառուցել գուշակ, որը մրցաշարի առաջին և երկրորդ տեղերը գրավողներին որոշող ցանկացած ալգորիթմում խաղերի արդյունքներն ընտրում է այնպես, որ մրցաշարի հաղթողն անցկացնի առնվազն $\lceil \log_2 n \rceil$ խաղ:

Գուշակի վարվելակերպը՝ խաղացողների հանդիպման արդյունքն ընտրելու կանոնը, սահմանենք հետևյալ եղանակով.

- *Պարտություն չունեցողը հաղթում է պարտություն ունեցողին:*
- *Պարտություն չունեցողներից հաղթում է նա, ով ավելի շատ հաղթանակ ունի:*
- *Մնացած դեպքերում ընտրվում է որևէ արդյունք, որը չի հակասում որևէ մասնակի կարգավորվածությանը:*

Դիտարկենք ավարտված մրցաշարի արդյունքները, որի խաղերի ելքերը կանխորոշված են սահմանված գուշակի կանոններին համաձայն: Կասենք, որ «a մասնակիցը գերազանցում է b մասնակցին» այն և միայն այն դեպքում, երբ $a = b$ կամ a մասնակիցը գերազանցում է այն խաղացողին, ով առաջինն է հաղթել b մասնակցին: Այդ առումով էական է խաղացողի միայն առաջին պարտությունը, իսկ նրա հետագա բոլոր խաղերն անտեսվում են: Համաձայն գուշակի

վարվելակերպի, յուրաքանչյուր մասնակից, ով առաջինն է հաղթել որևէ մեկին, իր անցկացրած խաղերից ոչ մեկում չպետք է ունենա պարտություն: Այստեղից հետևում է, որ այն մասնակիցը, ով հաղթել է իր առաջին p խաղերում, այդ խաղերի հիման վրա գերազանցում է ամենաշատը 2^p մասնակցի.

- Եթե $p = 0$, ապա դա ակնհայտ է, իսկ եթե $p > 0$, ապա p-րդ խաղը կայացել է այն մասնակցի հետ, ով կա՛մ ունեցել է պարտություն, կա՛մ էլ գերազանցում է առավելագույնը 2^{p-1} մասնակցի:

Մրցաշարի հաղթողը պետք է գերազանցի մրցաշարի բոլոր n մասնակիցներին, հետևաբար մրցաշարի հաղթողի անցկացրած խաղերի p քանակը պետք է բավարարի $n \le 2^p$ անհավասարությանը, որից հետևում է, որ $p \ge \lceil \log_2 n \rceil$: ■

§ 2.4. Մրցաշարի առաջին, երկրորդ և երրորդ տեղերը գրավողների որոշում

Դիտարկենք մրցաշար, որին մասնակցում են $n \ge 3$ խաղացողներ: Այդ մրցաշարի առաջին, երկրորդ և երրորդ տեղերը գրավողներին որոշող լավագույն ալգորիթմի բարդությունը նշանակենք $W_3(n)$:

2.9. ***Թեորեմ:*** $W_3(n) \le n - 3 + 2\lceil \log_2 n \rceil$:

Ապացույց: Առաջարկենք մրցաշարի առաջին, երկրորդ և երրորդ տեղերը գրավողների որոշման եղանակ: Մրցաշարի հաղթողին որոշենք գավաթային մրցակարգով ($n-1$ խաղ): Համաձայն 2.5. լեմմայի, երկրորդ տեղի հավակնորդների քանակն ամենաշատը $\lceil \log_2 n \rceil$ է: Նրանց համարակալենք $1, 2, \dots, \lceil \log_2 n \rceil$ թվերով հետևյալ կերպ. հավակնորդին վերագրենք k համարը, եթե նա մրցաշարի հաղթողին պարտվել է k-րդ փուլում (նկատենք, որ եթե հաղթողը մրցաշարը սկսել է երկրորդ փուլից, ապա 1 համարն ունեցող հավակնորդ չի լինի):

Հավակնորդներից հաղթողին որոշենք հետևյալ փոքր մրցաշարի միջոցով. Սկզբում մրցում են 1 և 2 համարներն ունեցող հավակնորդները, այնուհետև նրանց հաղթողի հետ մրցում է հերթական 3 համարի հավակնորդը և այդպես շարունակ:

Փոքր մրցաշարի խաղերի քանակը $\le \lceil \log_2 n \rceil - 1$ և այն որոշում է մրցաշարի երկրորդ տեղը գրավողին: Դժվար չէ նկատել, որ երկրորդ տեղը գրավողը հաղթել է ամենաշատը $\lceil \log_2 n \rceil$ խաղում: Իսկապես, դիցուք երկրորդ տեղը գրավողն եղել է k համարի հավակնորդը: Դա նշանակում է, որ սկզբնական գավաթային մրցակարգում նա հաղթել է ամենաշատը $\boldsymbol{k-1}$ խաղում, իսկ փոքր մրցաշարում՝ $\lceil \log_2 n \rceil - (k-1)$ խաղում: Հետևաբար մրցաշարի երկրորդ տեղը գրավողի հաղթանակների քանակը $\le \lceil \log_2 n \rceil$: Վերջինս նշանակում է, որ ամենաշատը $\lceil \log_2 n \rceil - 1$ խաղերի միջոցով՝ երկրորդ տեղը գրավողին պարտվողներից կորոշենք երրորդ տեղը գրավողին: Այսպիսով, առաջարկված եղանակով մրցաշարի առաջին, երկրորդ և երրորդ տեղերը գրավողներին կարող ենք որոշել առավելագույնը $n-3+2\lceil \log_2 n \rceil$ խաղերի միջոցով: ■

2.10. *Թեորեմ:* $W_3(n) \ge n-3+\lceil \log_2 n(n-1) \rceil$:

Ապացույց: Նախքան թեորեմի ապացույցին անդրադառնալը, ցույց տանք հետևյալ լեմմայի իսկությունը.

2.10a. *Լեմմա:* $\boldsymbol{n}$ տարբեր տարրերից՝ մաքսիմալ տարրի որոնման յուրաքանչյուր ծառ ունի առնվազն 2^{n-1} տերև:

Ապացույց: $n=1$ դեպքում պնդումն ակնհայտ է: Ենթադրենք, որ $n>1$ և լեմման տեղի ունի $(n-1)$ դեպքում: Դիտարկենք $\boldsymbol{n}$ տարբեր տարրերից՝ մաքսիմալ տարրն որոնող որևէ $\boldsymbol{T}$ ծառ: Ենթադրենք ծառի արմատում կատարվում է $a \,\vdots\, b$ համեմատությունը: Այդ դեպքում $a>b$ ելքին համապատասխան ենթածառում, ըստ ինդուկցիայի ենթադրության, կա առնվազն 2^{n-2} տերև (որովհետև $\boldsymbol{b}$ տարրը չի կարող լինել մաքսիմալ:) Նմանապես $a>b$ ելքին համապատասխան ենթածառում առկա է առնվազն 2^{n-2} տերև: Ուստի $\boldsymbol{T}$ ծառում տերևների քանակն առնվազն $2^{n-1}=(2^{n-2}+\ 2^{n-2})$ է: ■

Այժմ վերադառնանք թեորեմի ապացույցին: Դիցուք $a_1, a_2, \dots, a_n$ տարրերից (a_1, a_2) կարգավորված զույգն առաջին երկու տեղերը գրավողներն են: Դիտարկենք մրցաշարում առաջին երեք տեղերը գրավողներին որոշող որևէ ալգորիթմին համապատասխան $\boldsymbol{T}$ ծառը: Սահմանենք հետևյալ գուշակը. այն $\boldsymbol{T}$ ծառում համեմատությունների ելքերն ընտրում է այսպես.

$$a \,\vdots\, b = \begin{cases} a>b, & \text{եթե } a=a_1 \text{ և } b=a_2 \\ a>b, & \text{եթե } a\in\{a_1,a_2\} \text{ և } b\in\{a_3,\dots,a_n\}: \\ \textit{կամայական}, & \text{եթե } a,b\in\{a_3,\dots,a_n\} \end{cases}$$

Այդ դեպքում «**կամայական**» ելքով $a \,\vdots\, b$ տեսակի համեմատությունները $\boldsymbol{T}$ ծառում որոշում են $a_3, \dots, a_n$ տարրերից մաքսիմալ տարրն որոնող ալգորիթմին համապատասխան ենթածառ, որն, ըստ լեմմայի, ունի առնվազն 2^{n-3} տերև: Պարզ է, որ ամեն մի (a_{i_1}, a_{i_2}) կարգավորված զույգին, որտեղ $a_{i_1}, a_{i_2} \in \{a_1, a_2, \dots, a_n\}$ և $a_{i_1} \ne a_{i_2}$, համապատասխան ենթածառերի տերևների բազմությունները չեն հատվում: Իրարից տարբեր կարգավորված (a_{i_1}, a_{i_2}) զույգերի քանակը հավասար

է $n(n-1)$: Ուստի $\boldsymbol{T}$ ծառի տերևների քանակն առնվազն $2^{n-3}n(n-1)$, իսկ առավելագույնը՝ $2^{W_3(n)}$, այսինքն $2^{n-3}n(n-1) \le 2^{W_3(n)}$: Վերջինից էլ հետևում է $n-3+\lceil \log_2 n(n-1) \rceil \le W_3(n)$ անհավասարությունը: ■

Նշենք, որ 2.10. թեորեմի ապացույցի գաղափարը կարելի կիրառել

$$n-k+\left\lceil \log_2\big(n(n-1)(n-2)\cdots(n-k+2)\big) \right\rceil \le W_k(n)$$

գնահատականը ցույց տալու համար:

Ամփոփելով ստացված արդյունքները՝ կունենանք, որ

$$n-3+\lceil \log_2 n(n-1) \rceil \le W_3(n) \le n-3+2\lceil \log_2 n \rceil:$$

Նկատենք, որ այս խնդրի համար լավագույն ալգորիթմ չնշեցինք (այն հայտնի չէ), բայց գնահատեցինք նրա բարդությունը, ընդ որում վերևի և ներքևի գնահատականները կարող են տարբերվել ամենաշատը մեկով:

Գլուխ 3

ՏԵՍԱԿԱՎՈՐՄԱՆ ՀԱՄԱՐՅԱ ԼԱՎԱԳՈՒՅՆ ԱԼԳՈՐԻԹՄՆԵՐ

§ 3. 1. ՏԵՍԱԿԱՎՈՐՄԱՆ ԼԱՎԱԳՈՒՅՆ ԱԼԳՈՐԻԹՄԻ ԲԱՐԴՈՒԹՅԱՆ ՍՏՈՐԻՆ ԳՆԱՀԱՏԱԿԱՆԸ: ՏԵՍԱԿԱՎՈՐՈՒՄ ՏԵՂԱՎՈՐՄԱՆ ԵՂԱՆԱԿՈՎ

Այս բաժնում մենք կդիտարկենք **տեսակավորման խնդիրը** լուծող մի քանի ալգորիթմներ: Այդ խնդիրը կարելի է ձևակերպել այսպես. Դիցուք տրված են զույգ առ զույգ տարբեր քաշերով $\boldsymbol{a_1, a_2, \dots, a_n}$ միանման կշռաքարերը, որոնց քաշերը, կշռաքարերին նայելով, հնարավոր չէ համեմատել: Տրված է նաև նժարավոր կշեռք, որի միջոցով կարող ենք համեմատել ընտրված երկու կշռաքարերի քաշերը: Պահանջվում է նվազագույն թվով կշռումների արդյունքում կշռաքարերը դասավորել ըստ նրանց քաշերի աճման կամ նվազման կարգի:

Համեմատությունների միջոցով տեսակավորման խնդիրը կարելի է ձևակերպել նաև այլ կերպ: Օրինակ, եթե որևէ մրցաշարում մասնակցում են $\boldsymbol{n}$ թվով խաղացող, ապա հարկավոր է հնարավորինս քիչ քանակի խաղերի կազմակերպման արդյունքում խաղացողներին դասավորել ըստ նրանց ուժերի աճման կամ նվազման կարգի` այն ենթադրությամբ, որ հավասար ուժի խաղացողներ չկան:

Հեշտ է նկատել, որ տեսակավորման խնդիրը հանդիսանում է որոնման տիպի խնդիր. $\boldsymbol{n}!$ հատ տեղադրությունից անհրաժեշտ է գտնել որոնելի դասավորմանը համապատասխանող տեղադրությունը` կատարելով նվազագույն թվով համեմատություններ:

Դիցուք $\boldsymbol{S(n)}$ հանդիսանում է համեմատությունների այն նվազագույն քանակը, որն անհրաժեշտ է $\boldsymbol{n}$ տարրերը վատագույն դեպքում տեսակավորելու համար: Այդ պարագայում $\boldsymbol{S(n)}$ մեծության համար ճշմարիտ է հետևյալ ստորին գնահատականը.

3. 1. ***Թեորեմ:*** $\lceil \log_2 \boldsymbol{n}! \rceil \le \boldsymbol{S(n)}$ ։

Ապացույց: Տեսակավորման խնդիրը լուծող ցանկացած ալգորիթմին համապատասխան **2**-ծառի բարձրությունն առնվազն $\boldsymbol{S(n)}$ է, իսկ այդ ծառի եզրային գագաթների (տերևների) քանակը` առնվազն $\boldsymbol{n}!$ ։ Ուստի

$$\boldsymbol{n}! \le \boldsymbol{2^{S(n)}} \;\Rightarrow\; \log_2 \boldsymbol{n}! \le \boldsymbol{S(n)}$$

և, քանի որ $\boldsymbol{S(n)}$ հանդիսանում է ամբողջ թիվ, ապա $\lceil \log_2 \boldsymbol{n}! \rceil \le \boldsymbol{S(n)}$: ■

3. 2. ***Սահմանում:*** *Դիցուք* $\boldsymbol{a_1, a_2, \dots, a_n}$ *տարրերը տեսակավորող որևէ ալգորիթմի բարդությունը* $\boldsymbol{T(n)}$ *է: Այդ դեպքում ընդունված է ասել, որ այդ ալգորիթմը հանդիսանում է* ***տեսակավորման համարյա լավագույն ալգորիթմ****, եթե* $\boldsymbol{lim_{n\to\infty} \frac{T(n)}{S(n)} = 1}$:

Համաձայն Ստիռլինգի $\boldsymbol{n}! \sim \left(\frac{\boldsymbol{n}}{\boldsymbol{e}}\right)^{\boldsymbol{n}} \sqrt{\boldsymbol{2\pi n}}$ բանաձևի ունենք, որ $\log_2 \boldsymbol{n}! \sim \boldsymbol{n} \log_2 \boldsymbol{n}$: Հետևաբար, հաշվի առնելով

$$\boldsymbol{1} \le \frac{\boldsymbol{T(n)}}{\boldsymbol{S(n)}} \le \frac{\boldsymbol{T(n)}}{\lceil \log_2 \boldsymbol{n}! \rceil}$$

անհավասարությունները, կարող ենք պնդել, որ եթե $\boldsymbol{T(n)}$ բարդությամբ տեսակավորման ալգորիթմի համար $\boldsymbol{T(n)} \sim \boldsymbol{n} \log_2 \boldsymbol{n}$, ապա այդ ալգորիթմը տեսակավորման համարյա լավագույն ալգորիթմ է:

Այժմ դիտարկենք տեսակավորված շարքում տարրի տեղավորման օժանդակ խնդիրը, որից հետո կնկարագրենք տեղավորման եղանակով տեսակավորման ալգորիթմը: Ենթադրենք տրված է $\boldsymbol{k}$ տարրերի տեսակավորված $\boldsymbol{a_1 < a_2 < \dots < a_k}$ շարքը և անհրաժեշտ է նվազագույն թվով համեմատությունների միջոցով այդ շարքում տեղավորել $\boldsymbol{a}$ տարրը: Վերջինս նշված շարքում կարող է զբաղեցնել հնարավոր $(\boldsymbol{k+1})$ տեղերից որևէ մեկը: Պարզ է, որ $(\boldsymbol{k+1})$ հնարավորություններից ճիշտ մեկն որոշող ալգորիթմի բարդությունն առնվազն $\lceil \log_2(\boldsymbol{k+1}) \rceil$ է: Մյուս կողմից **1. 2.** ալգորիթմը, որը յուրաքանչյուր քայլում հնարավորությունների բազմությունը տրոհում է համարյա հավասար մասերի, առաջարկված խնդիրը կլուծի առավելագույնը $\lceil \log_2(\boldsymbol{k+1}) \rceil$ համեմատությունների միջոցով: Այսպիսով առաջարկվեց $\boldsymbol{k}$ տարրերի տեսակավորված շարքում նոր տարրի տեղավորման խնդիրը լուծող լավագույն ալգորիթմ, որի բարդությունը հավասար է $\lceil \log_2(\boldsymbol{k+1}) \rceil$:

3. 3. *Ալգորիթմ (տեսակավորում տեղավորման եղանակով):*

Ալգորիթմը հերթական $\boldsymbol{k}$*-րդ քայլում* $\boldsymbol{a_k}$ *տարրը տեղավորում է* $\boldsymbol{a_1, a_2, \dots, a_{k-1}}$ *տարրերից կազմված* $\boldsymbol{X}$ *տեսակավորված շարքում:*

(i) *Ընդունել* $\boldsymbol{X = a_1}$ *և* $\boldsymbol{k = 2}$:
(ii) *Օգտագործելով կիսումների եղանակը՝* $\boldsymbol{a_k}$ *տարրը տեղավորել* $\boldsymbol{X}$ *շարքում և* $\boldsymbol{k := k + 1}$:
(iii) *Եթե* $\boldsymbol{k < n}$*, ապա վերադառնալ* (ii) *քայլի կատարմանը, հակառակ դեպքում՝ ավարտել ալգորիթմի աշխատանքը:*

Հեշտ է համոզվել, որ նկարագրված 3.3. ալգորիթմը տեսակավորում է $a_1, a_2, \dots, a_n$ տարրերը՝ կատարելով

$$A(n) = \lceil \log_2 2 \rceil + \lceil \log_2 3 \rceil + \dots + \lceil \log_2 n \rceil = \sum_{k=2}^{n} \lceil \log_2 k \rceil$$

համեմատություն: Քանի որ $2^{i-1} < k \le 2^i$ դեպքում $\lceil \log_2 k \rceil = i$ և $2^{\lceil \log_2 n \rceil - 1} < n \le 2^{\lceil \log_2 n \rceil}$, ապա

$$A(n) = 1 + 2 \cdot 2^1 + 3 \cdot 2^2 + \dots + (\lceil \log_2 n \rceil - 1) \cdot 2^{\lceil \log_2 n \rceil - 2} + \lceil \log_2 n \rceil \cdot \left(n - 2^{\lceil \log_2 n \rceil - 1}\right):$$

Այնուհետև, օգտագործելով

$$(1 + 2 \cdot x^1 + 3 \cdot x^2 + \dots + m \cdot x^{m-1})(x - 1) = mx^m - (1 + x + x^2 + \dots + x^{m-1}) \Rightarrow$$

$$1 + 2 \cdot x^1 + 3 \cdot x^2 + \dots + m \cdot x^{m-1} = \frac{mx^m}{(x-1)} - \frac{x^m - 1}{(x-1)^2}$$

հավասարությունները $x = 2$ և $m = \lceil \log_2 n \rceil - 1$ դեպքում, ստանում ենք

$$A(n) = (\lceil \log_2 n \rceil - 1) \cdot 2^{\lceil \log_2 n \rceil - 1} - 2^{\lceil \log_2 n \rceil - 1} + 1 + \lceil \log_2 n \rceil \cdot \left(n - 2^{\lceil \log_2 n \rceil - 1}\right) =$$

$$= n \lceil \log_2 n \rceil - 2^{\lceil \log_2 n \rceil} + 1:$$

Եվ քանի որ $A(n) \sim n \log_2 n$, ապա տեղավորման եղանակով տեսակավորման ալգորիթմը համարյա լավագույն ալգորիթմ է:

§ 3. 2. ՏԵՍԱԿԱՎՈՐՈՒՄ ՇԱՐՔԵՐԻ ՁՈՒԼՄԱՆ ԵՂԱՆԱԿՈՎ

Արդ դիտարկենք հետևյալ հարցը. Ո՞րն է տրված երկու՝ $a_1 < a_2 < \dots < a_n$ և $b_1 < b_2 < \dots < b_m$ կարգավորված շարքերի ձուլման լավագույն եղանակը, երբ թույլատրվում է կատարել $a_i \vdots b_j$ տեսակի համեմատություններ: $M(n, m)$ գրությամբ նշանակենք համեմատությունների այն նվազագույն քանակը, որն անհրաժեշտ է n և m երկարություններով կարգավորված շարքերը ձուլելու համար:

3.4. *Թեորեմ:* $\left\lceil \log_2 \binom{n+m}{n} \right\rceil \le M(n, m) \le n + m - 1$:

Ապացույց: Կարգավորված $a_1 < a_2 < \dots < a_n$ և $b_1 < b_2 < \dots < b_m$ շարքերը ձուլող ցանկացած ալգորիթմին համապատասխան 2-ծառի բարձրությունն առնվազն $M(n, m)$ է, իսկ այդ ծառի եզրային գագաթների (տերևների) քանակը՝ առնվազն $\binom{n+m}{n}$, քանի որ ստացվող $n + m$ երկարությամբ շարքում $a_1, a_2, \dots, a_n$ տարրերը կարող են զբաղեցնել ցանկացած n տեղեր: Ուստի

$$\binom{n+m}{n} \le 2^{M(n,m)} \;\Rightarrow\; \log_2 \binom{n+m}{n} \le M(n, m)$$

և, քանի որ $M(n, m)$ հանդիսանում է ամբողջ թիվ, ապա $\left\lceil \log_2 \binom{n+m}{n} \right\rceil \le M(n, m)$:

Վերին գնահատականի իսկությունն ապացուցելու համար առաջարկենք $a_1 < a_2 < \dots < a_n$ և $b_1 < b_2 < \dots < b_m$ կարգավորված շարքերը ձուլող և $n + m - 1$ բարդությամբ ալգորիթմ:

3.5. *Ալգորիթմ (շարքերի ձուլման պարզագույն ալգորիթմ):*

(i) *Որպես X շարք ընդունել $a_1 < a_2 < \dots < a_n$ շարքը, իսկ որպես Y շարք՝ $b_1 < b_2 < \dots < b_m$ շարքը: Z շարքը դատարկ է:*

(ii) *Համեմատել X և Y շարքերի ամենափոքր տարրերը: Նրանցից ամենափոքրը հանել համապատասխան շարքից և տեղադրել Z շարքի հերթական տեղում:*

(iii) *Եթե X և Y շարքերից մեկը դատարկ է, ապա մյուս շարքն աջից կցագրել Z շարքին և ավարտել ալգորիթմի աշխատանքը, հակառակ դեպքում՝ վերադառնալ* (ii) *քայլի կատարմանը:*

Հեշտ է ստուգել, որ շարքերի ձուլման պարզագույն ալգորիթմի բարդությունը հավասար է $n + m - 1$: ■

Թեորեմում նշված վերին և ստորին գնահատականները միմյանցից էապես տարբերվում են, իսկ $M(n, m)$ մեծությունը կարող է մոտ լինել նրանցից յուրաքանչյուրին: Այսպես, նախորդ պարագրաֆում մենք ցույց ենք տվել, որ $M(1, m) = \lceil \log_2(m + 1) \rceil$, սակայն շարքերի ձուլման պարզագույն ալգորիթմի բարդությունն այդ դեպքում հավասար է m և, պարզ է, որ նրա օգտագործումը նպատակահարմար չէ: Պարզագույն ալգորիթմը հարմար է կիրառել այն դեպքում, երբ m ու n արժեքներն իրարից քիչ են տարբերվում: Մասնավորապես ճշմարիտ է հետևյալ արդյունքը.

3.6. *Թեորեմ:* Յուրաքանչյուր n բնական թվի համար $M(n, n) = 2n - 1$, այսինքն հավասար երկարություն ունեցող կարգավորված շարքերի դեպքում՝ ձուլման պարզագույն ալգորիթմը լավագույնն է:

Ապացույց: Դիտարկենք $a_1 < a_2 < \dots < a_n$ և $b_1 < b_2 < \dots < b_m$ շարքերը ձուլող ցանկացած ալգորիթմ: Նկարագրենք $a_i \vdots b_j$ համեմատության արդյունքն որոշող «գուշակին».

$$a_i \vdots b_j = \begin{cases} a_i < b_j, & \text{երբ } i < j \\ a_i > b_j, & \text{երբ } i \ge j \end{cases}:$$

Հասկանալի է, որ սահմանված գուշակը, շարքերի ձուլման ցանկացած ալգորիթմում, ընտրում է որոշակի ճյուղ, որին համապատասխանում է a_i և b_j տարրերի

$$b_1 < a_1 < b_2 < a_2 < b_3 < a_3 < \dots < b_n < a_n$$

կարգավորվածությունը: Այդ դեպքում պարտադիր կատարվել են

$$(\boldsymbol{b_1} \vdots \boldsymbol{a_1}), (\boldsymbol{a_1} \vdots \boldsymbol{b_2}), (\boldsymbol{b_2} \vdots \boldsymbol{a_2}), (\boldsymbol{a_2} \vdots \boldsymbol{b_3}), \dots, (\boldsymbol{b_n} \vdots \boldsymbol{a_n})$$

համեմատություններից յուրաքանչյուրը (առնվազն $(\boldsymbol{2n-1})$ համեմատություն), քանի որ եթե, մասնավորապես, չի կատարվել $(\boldsymbol{a_1} \vdots \boldsymbol{b_2})$ համեմատությունը, ապա գուշակի ընտրած ճյուղին կբավարարի նաև

$$\boldsymbol{b_1 < b_2 < a_1 < a_2 < b_3 < a_3 < \dots < b_n < a_n}$$

կարգավորվածությունը և, հետևաբար, շարքերը ձուլող ալգորիթմը չի կարող տարբերակել նշված երկու կարգավորվածությունները: ■

Այս ապացույցի պարզ ձևափոխությունը հաստատում է $\boldsymbol{M(n, n+1) = 2n}$ հավասարությունը բոլոր $\boldsymbol{n \geq 0}$ համար:

Այժմ նկարագրենք $\boldsymbol{a_1, a_2, \dots, a_n}$ տարրերը տեսակավորող ալգորիթմ, որը հիմնվում է շարքերի ձուլման պարզագույն ալգորիթմի վրա:

3.7.1. *Ալգորիթմ (տեսակավորում շարքերի ձուլման եղանակով):*

$\boldsymbol{n = 2^k}$ *դեպք:*

(i) *Որպես ձուլման ենթակա շարքեր ընդունել* $\boldsymbol{a_1, a_2, \dots, a_n}$ *տարրերը:*

(ii) *Ձուլման ենթակա շարքերը տրոհել զույգերի և շարքերի ձուլման պարզագույն ալգորիթմի միջոցով՝ յուրաքանչյուր զույգից ստանալ ձուլման ենթակա մեկ շարք:*

(iii) *Եթե ձուլման ենթակա շարքերի քանակը մեկ է, ապա ավարտել ալգորիթմի աշխատանքը, հակառակ դեպքում՝ վերադառնալ* (ii) *քայլի կատարմանը:*

Հեշտ է համոզվել, որ $\boldsymbol{i}$-րդ $(\boldsymbol{i = 1, 2, \dots, k})$ ձուլումից հետո ստացվում են $\boldsymbol{2^{k-i}}$ հատ ձուլման ենթակա շարք, որոնցից յուրաքանչյուրի երկարությունը $\boldsymbol{2^i}$ է: Ուստի $\boldsymbol{i}$-րդ ձուլման ընթացքում կատարվող գործողությունների քանակը կլինի $\boldsymbol{2^{k-i}(2^i-1)}$, իսկ 3.7.1. ալգորիթմի $\boldsymbol{B(n = 2^k)}$ բարդությունը՝

$$\boldsymbol{B(n = 2^k) = \sum_{i=1}^{k} 2^{k-i}(2^i - 1) = k \cdot 2^k - 2^k + 1}:$$

3.7.2. *Ալգորիթմ (տեսակավորում շարքերի ձուլման եղանակով):*

Ընդհանուր դեպք:

(i) $\boldsymbol{n}$ *թիվը ներկայացնել թվարկության երկուական համակարգում՝* $\boldsymbol{n = 2^{k_1} + 2^{k_2} + \dots + 2^{k_s}}$*, որտեղ* $\boldsymbol{k_1 > k_2 > \dots > k_s \geq 0}$*, և օգտագործելով* $\boldsymbol{2^k}$ *դեպքում տեսակավորման* 3.7.1. *ալգորիթմը՝* $\boldsymbol{a_1, a_2, \dots, a_n}$ *տարրերից կառուցել* $\boldsymbol{2^{k_1}, 2^{k_2}, \dots, 2^{k_s}}$ *երկարություններով շարքեր:*

(ii) *Ստացված շարքերից հերթականորեն ընտրել ամենակարճ երկարությամբ երկու շարք և նրանց ձուլել շարքերի ձուլման պարզագույն ալգորիթմի միջոցով՝ մինչև մեկ շարքի ստանալը:*

Հասկանալի է, որ $\boldsymbol{2^{k_1}, 2^{k_2}, \dots, 2^{k_s}}$ երկարություններով շարքերը ստանալու համար կպահանջվի $\sum_{i=1}^{s}\boldsymbol{(k_i \cdot 2^{k_i} - 2^{k_i} + 1)}$ համեմատություն, իսկ երկրորդ քայլում $\boldsymbol{2^{k_1} + 2^{k_2} + \dots + 2^{k_s}}$ երկարությամբ շարք ստանալու համար կպահանջվի

$$\boldsymbol{(2^{k_{s-1}} + 2^{k_s} - 1) + (2^{k_{s-2}} + 2^{k_{s-1}} + 2^{k_s} - 1) + \dots + (2^{k_1} + 2^{k_2} + \dots + 2^{k_s} - 1) =}$$
$$\boldsymbol{= \sum_{i=1}^{s-1}(2^{k_i} + 2^{k_{i+1}} + \dots + 2^{k_s} - 1)}$$

համեմատություն: Հետևաբար 3.7.2. ալգորիթմի բարդությունը կլինի

$$\boldsymbol{B(n) = \sum_{i=1}^{s}(k_i \cdot 2^{k_i} - 2^{k_i} + 1) + \sum_{i=1}^{s-1}(2^{k_i} + 2^{k_{i+1}} + \dots + 2^{k_s} - 1) =}$$
$$\boldsymbol{= 1 - 2^{k_s} + \sum_{i=1}^{s}(k_i + i - 1) \cdot 2^{k_i} \sim n \log_2 n,}$$

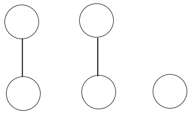
որից հետևում է, որ ձուլման եղանակով տեսակավորման ալգորիթմը համարյա լավագույն ալգորիթմ է:

Նշենք, որ $\boldsymbol{A(n) \leq B(n)}$ և, չնայած այս փաստին, մենք նկարագրեցինք ձուլման եղանակով տեսակավորման ալգորիթմը, քանի որ այն հեշտ իրացվող է և ունի գործնական նշանակություն:

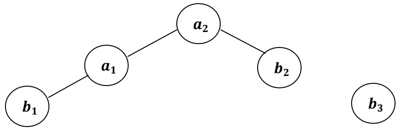
§ 3.3. ՏԵՍԱԿԱՎՈՐՈՒՄ ՁՈՒԼՄԱՆ ԵՎ ՏԵՂԱՎՈՐՄԱՆ ԵՂԱՆԱԿՈՎ

Նախորդ երկու՝ տեղավորման եղանակով տեսակավորման և շարքերի ձուլման եղանակով տեսակավորման ալգորիթմների նկարագրություններից հետևում է, որ $\boldsymbol{S(n) = A(n) = B(n) = \lceil \log_2 n! \rceil}$, երբ $\boldsymbol{n = 1, 2, 3, 4}$:

Դիտարկենք $\boldsymbol{n = 5}$ դեպքը: Հեշտ է համոզվել, որ $\boldsymbol{A(5) = 8}$, $\boldsymbol{B(5) = 9}$ և $\boldsymbol{\lceil \log_2 5! \rceil = 7}$: Ցույց տանք, որ $\boldsymbol{S(5) = 7}$: Առանձնացնենք տեսակավորվող հինգ տարրերից որևէ մեկը, իսկ մնացածներին տրոհենք զույգերի և կարգավորենք յուրաքանչյուր զույգ (նկ. 3.1), որից հետո կարգավորենք նաև զույգերի մեծ տարրերը (նկ. 3.2):



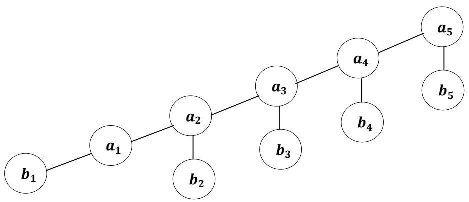
Նկար 3.1:



Նկար 3.2:

Այնուհետև, եթե b_2 տարրը տեղավորենք $b_1 < a_1 < a_2$ շարքում (վատագույն դեպքում կպահանջվի **2** համեմատություն), որից հետո b_3 տարրը տեղավորենք ստացված շարքում (վատագույն դեպքում կպահանջվի **3** համեմատություն), ապա **5** տարրերը կտեսակավորվեն **8** համեմատությունների արդյունքում: Սակայն կարելի է տեղավորվող b_2 և b_3 տարրերի հերթականությունը փոխարինել b_3, b_2 հերթականությամբ. սկզբում b_3 տարրը տեղավորել $b_1 < a_1 < a_2$ շարքում (կատարելով ամենաշատը **2** համեմատություն), իսկ այնուհետև, հաշվի առնելով $b_2 < a_2$ պայմանը, ստացված շարքում տեղավորել b_2 տարրը (կատարելով առավելագույնն ևս **2** համեմատություն): Արդյունքում **5** տարրերը կտեսակավորվեն **7** համեմատությունների կատարմամբ: Հետևաբար $S(5) = 7$:

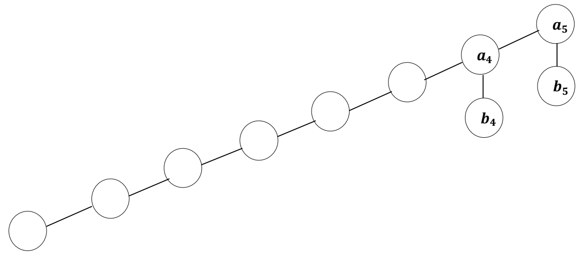
Օգտագործելով **5** տարրերի վերը նկարագրված եղանակը՝ տեսակավորենք **10** տարրերը: Նախ այդ տարրերը տրոհենք զույգերի և յուրաքանչյուր զույգ կարգավորենք (**5** համեմատություն): Հաջորդիվ, տեսակավորենք զույգերի մեծ տարրերն ըստ վերը նշված ալգորիթմի (առավելագույնը **7** համեմատություն): Արդյունքում կունենանք 3.3 նկարում ներկայացված պատկերը:



Նկար 3.3:

Տեսակավորումն ավարտելու համար բավական է b_2, b_3, b_4, b_5 տարրերը տեղավորել $b_1 < a_1 < a_2 < a_3 < a_4 < a_5$ շարքում: Տեղավորումն իրականացնենք b_3, b_2, b_5, b_4 հերթականությամբ: Առաջին երկու՝ b_3, b_2 տարրերից յուրաքանչյուրի

տեղավորման համար կկատարենք առավելագույնը **2** համեմատություն և կունենանք հետևյալ



Նկար 3.4:

պատկերը: Այնուհետև ստացված շարքում տեղավորենք b_5 և b_4 տարրերը՝ յուրաքանչյուրի տեղավորման համար կատարելով առավելագույնը **3** համեմատություն: Արդյունքում **10** տարրերի տեսակավորման համար առավելագույնը կկատարվի $5 + 7 + 2 \cdot 2 + 2 \cdot 3 = 22$ համեմատություն: Ուստի $22 = \lceil \log_2 10! \rceil \le S(10) \le 22 \Rightarrow S(10) = 22$:

Այժմ ընդհանրացնենք **5** և **10** տարրերի տեսակավորման՝ վերը նկարագրված ալգորիթմները:

3.8. ***Ալգորիթմ (տեսակավորում շարքերի ձուլման և տեղավորման եղանակով):***

(i) *n տարրերը տրոհել զույգերի ($n \equiv 1 (mod 2)$ դեպքում առանձնացնել ճիշտ մեկ տարր) և յուրաքանչյուր զույգ կարգավորել (ընդամենը $\left\lfloor \frac{n}{2} \right\rfloor$ զույգ):*

(ii) *Օգտվելով տեսակավորման ձուլման և տեղավորման ալգորիթմից՝ տեսակավորել $\left\lfloor \frac{n}{2} \right\rfloor$ զույգերի մեծ տարրերը: Զույգերի՝ տեսակավորված մեծ տարրերը նշանակել $a_1 < a_2 < \dots < a_{\lfloor \frac{n}{2} \rfloor}$, իսկ փոքր տարրերը՝ համապատասխանաբար $b_1, b_2, \dots, b_{\lceil \frac{n}{2} \rceil}$ ($n \equiv 1 (mod 2)$ դեպքում առանձնացված տարրը նշանակել $b_{\lceil \frac{n}{2} \rceil}$): Գլխավոր շղթա անվանել $b_1 < a_1 < a_2 < \dots < a_{\lfloor \frac{n}{2} \rfloor}$ շարքը:*

(iii) *Կիսումների եղանակով $b_2, b_3, \dots, b_{\lceil \frac{n}{2} \rceil}$ տարրերը տեղավորել գլխավոր շղթայում*

$$b_3, b_2;\ b_5, b_4;\ b_{11}, b_{10}, b_9, b_8, b_7, b_6;\ \dots;\ b_{t_k}, b_{t_k-1}, \dots, b_{t_{k-1}+1};\ \dots$$

հերթականությամբ, որտեղ t_k փոփոխականը ցույց է տալիս այն ամենամեծ համարը, որ b_{t_k} տարրը գլխավոր շղթայում հնարավոր է տեղավորել k համեմատությունների արդունքում, երբ շղթայում արդեն տեղավորված են $b_2, b_3, \dots, b_{t_{k-1}}$ տարրերը: Սրանով ալգորիթմի նկարագիրն ավարտված է:

Նշված ալգորիթմի երրորդ քայլի ճշգրիտ կատարման համար հարկավոր է որոշել $(t_2, t_3, \dots, t_k, \dots)$ հաջորդականության ընդհանուր անդամի արժեքը: Դժվար չէ համոզվել, որ $t_2 = 3$ և $t_3 = 5$: Նկատենք, որ եթե գլխավոր շղթայում արդեն դասավորված են $b_2, b_3, \dots, b_{t_{k-1}}$ տարրերը, ապա այդ շղթայում a_{t_k} տարրից փոքր են միայն $a_1, a_2, \dots, a_{t_k-1}$ և $b_1, b_2, \dots, b_{t_{k-1}}$ տարրերը, հետևաբար, b_{t_k} տարրն այդ շղթայում կարող է զբաղեցնել $t_{k-1} + t_k$ տեղերից որևէ մեկը և, որպեսզի t_k արժեքը լինի ամենամեծ համարը, որի դեպքում b_{t_k} տարրը հնարավոր է տեղավորել գլխավոր շղթայում՝ կատարելով առավելագույնը k համեմատություն, պետք է տեղի ունենա

$$t_{k-1} + t_k = 2^k$$

հավասարությունը: Ստացված անդրադարձ առնչությունը և $t_2 = 3$ սկզբնական արժեքը միարժեքորեն որոշում են $(t_2, t_3, \dots, t_k, \dots)$ հաջորդականությունը: Նկատենք, որ $t_1 = 1$ սկզբնական արժեքի դեպքում ևս կստացվի նույն հաջորդականությունը: Ուստի

$$\begin{array}{r|c|}
t_1 = 1 & (-1)^1 \\
t_1 + t_2 = 2^2 & (-1)^2 \\
t_2 + t_3 = 2^3 & (-1)^3 \\
\vdots & \vdots \\
t_{k-1} + t_k = 2^k & (-1)^k
\end{array}$$

հավասարություններից ստանում ենք

$$(-1)^k t_k = (-1)^1 + (-2)^2 + (-2)^3 + \dots + (-2)^k =$$

$$= (-2)^0 + (-2)^1 + (-2)^2 + (-2)^3 + \dots + (-2)^k = \frac{(-2)^{k+1} - 1}{-2 - 1} = \frac{(-1)^k \cdot 2^{k+1} + 1}{3} \Rightarrow$$

$$\Rightarrow t_k = \frac{2^{k+1} + (-1)^k}{3}:$$

Պարզ է, որ ըստ ստացված բանաձևի $t_0 = 1$:

Հաջորդիվ, տեսակավորման ձուլման և տեղավորման ալգորիթմի բարդությունը նշանակենք $F(n)$, իսկ այդ ալգորիթմի երրորդ քայլում $b_2, b_3, \dots, b_m$ տարրերը գլխավոր շղթայում տեղավորելու համար անհրաժեշտ համեմատությունների քանակը՝ $G(m)$: Հեշտ է համոզվել, որ այդ դեպքում

$$F(n) = \left\lfloor \frac{n}{2} \right\rfloor + F\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + G\left(\left\lceil \frac{n}{2} \right\rceil\right),$$

$$G(m) = \sum_{j=1}^{k-1} j\left(t_j - t_{j-1}\right) + k(m - t_{k-1}),$$

երբ $t_{k-1} < m \le t_k$: Պարզեցնենք $G(m)$ մեծությունը հետևյալ կերպ.

$$G(m) = 1 \cdot (t_1 - t_0) + 2 \cdot (t_2 - t_1) + \dots + (k-1)(t_{k-1} - t_{k-2}) + k(m - t_{k-1}) =$$

$$= km - (t_0 + t_1 + t_2 + \dots + t_{k-1}) = km - \sum_{j=1}^{k-1} \left(\frac{2^{j+1} + (-1)^j}{3}\right) = km - \left\lfloor \frac{2^{k+1}}{3} \right\rfloor:$$

Ալգորիթմի $F(n)$ բարդության տեսքը պարզեցնելու համար ապացուցենք հետևյալ օժանդակ արդյունքը.

3.9. *Լեմմա: Եթե* $\frac{2^{k+1}}{3} < n < \frac{2^{k+2}}{3}$, *ապա* $F(n) - F(n-1) = k$:

Ապացույց: Նախ նկատենք, որ $F(1) = 0$, $F(2) = 1$, $F(3) = 3$, $F(4) = 5$ և $F(5) = 7$: Երբ $k = 1$, ապա $n = 2$ և $F(2) - F(1) = 1$: Իսկ երբ $k = 2$, ապա $3 \le n \le 5$ և $F(5) - F(4) = F(4) - F(3) = F(3) - F(2) = 2$:

Այսպիսով լեմման ստույգ է $2 \le n \le 5$ դեպքում: Ենթադրենք լեմման ճշմարիտ է, երբ տեսակավորվող տարրերի քանակը փոքր է n թվից և ապացուցենք, որ լեմման տեղի ունի նաև $\boldsymbol{n}$ տարրերի դեպքում: Առանձին-առանձին դիտարկենք $\boldsymbol{n \equiv 0(mod2)}$ և $\boldsymbol{n \equiv 1(mod2)}$ դեպքերը: Դիցուք $\boldsymbol{n = 2p}$ և $\frac{2^{k+1}}{3} < \boldsymbol{2p} < \frac{2^{k+2}}{3}$: Այդ դեպքում $\frac{2^k}{3} < \boldsymbol{p} < \frac{2^{k+1}}{3}$ և

$$\boldsymbol{F(2p) - F(2p-1) = [p + F(p) + G(p)] - [(p-1) + F(p-1) + G(p)] =}$$

$$\boldsymbol{= F(p) - F(p-1) + 1} \underbrace{=}_{\text{ըստ ինդուկցիայի}} \boldsymbol{k - 1 + 1 = k}:$$

Իսկ երբ $\boldsymbol{n = 2p + 1}$ և $\frac{2^{k+1}}{3} < \boldsymbol{2p + 1} < \frac{2^{k+2}}{3}$, ապա ունենք, որ

$$\boldsymbol{\frac{2^{k+1}}{3} + 1 < 2p + 2 < \frac{2^{k+2}}{3} + 1 \;\Rightarrow\; \frac{2^k}{3} + \frac{1}{2} < p + 1 < \frac{2^{k+1}}{3} + \frac{1}{2} \;\Rightarrow}$$

$$\boldsymbol{\frac{2^k}{3} + \frac{(-1)^{k-1}}{3} < \frac{2^k}{3} + \frac{1}{2} < p + 1 < \frac{2^{k+1}}{3} + \frac{1}{2} = \frac{2^{k+1}}{3} + \frac{(-1)^k}{3} + \frac{1}{2} - \frac{(-1)^k}{3}}$$

$$\boldsymbol{\Rightarrow\; t_{k-1} < p + 1 \leq t_k} \text{ և}$$

$$\boldsymbol{F(2p+1) - F(2p) = [p + F(p) + G(p+1)] - [p + F(p) + G(p)] = G(p+1) - G(p)}:$$

Քանի որ $\boldsymbol{G(p+1) - G(p)}$ տարբերությունը հանդիսանում է $\boldsymbol{b_{p+1}}$ տարրը գլխավոր շղթայում տեղավորելու համար անհրաժեշտ համեմատությունների քանակը, իսկ $\boldsymbol{t_{k-1} < p + 1 \leq t_k}$, ապա $\boldsymbol{F(2p+1) - F(2p) = G(p+1) - G(p) = k}$: ■

Պարզ է, որ $\frac{2^{k+1}}{3} < \boldsymbol{n} < \frac{2^{k+2}}{3}$ պայմանը համարժեք է $\boldsymbol{k = \left\lceil \log_2 \frac{3}{4} n \right\rceil}$ հավասարությանը, այսինքն, ըստ ապացուցված լեմմայի, ճշմարիտ է

$$\boldsymbol{F(n) - F(n-1) = \left\lceil \log_2 \frac{3}{4} n \right\rceil}$$

անդրադարձ առնչությունը, որն էլ լուծելով՝ ստանում ենք, որ

$$\boldsymbol{F(n) = \left\lceil \log_2 \frac{3}{4} 2 \right\rceil + \left\lceil \log_2 \frac{3}{4} 3 \right\rceil + \cdots + \left\lceil \log_2 \frac{3}{4} n \right\rceil = \sum_{k=2}^{n} \left\lceil \log_2 \frac{3}{4} k \right\rceil}:$$

Եվ քանի որ $\boldsymbol{F(n) \leq A(n)}$, ապա ձուլման և տեղավորման եղանակով տեսակավորման ալգորիթմը համարյա լավագույն ալգորիթմ է: Անմիջական

հաշվարկի միջոցով կարելի է համոզվել, որ երբ $\boldsymbol{1 \leq n \leq 11}$ կամ $\boldsymbol{20 \leq n \leq 21}$, ապա $\boldsymbol{F(n) = S(n) = \lceil \log_2 n! \rceil}$:

Մարկ Ուէլսի կողմից Մանիակ II համակարգչի վրա իրականացվել է սպառիչ որոնում (մոտավորապես $\boldsymbol{60}$ ժամ), որը ցույց է տվել, որ $\boldsymbol{S(12) = F(12) = 30}$ [Proc. IFIP Congress 65, 2 (1965), 497 – 498]:

Մինչև հիմա հայտնի չէ $\boldsymbol{n}$ փոփոխականի որևէ արժեք, որի համար տեղի ունենա $\boldsymbol{S(n) < F(n)}$ անհավասարությունը: