DOI: 10.65322/j.srd.03172026.08

ALGORITMLARNI LOYIHALASHTIRISH VA TAHLIL QILISH

Sayidov N.A., *PhD, dotsent,*
E-mail: sayidovnozimjon@gmail.com,
Sabirov B.T., *magistrant,*
University of business and science
Xusanov S.X., *talaba,*
Namangan state technical university

Annotatsiya: *Maqolada algoritmlarni loyihalashtirish va ularni tahlil qilishning nazariy hamda amaliy jihatlari yoritilgan. Algoritmlarni ishlab chiqishda samaradorlik, aniqlik va resurslardan oqilona foydalanish asosiy mezon sifatida ko‘rib chiqiladi. Shuningdek, algoritmlarni baholashning asosiy usullari, jumladan vaqt va xotira murakkabligi, Big-O notatsiyasi, hamda zamonaviy algoritmik yondashuvlar tahlil qilingan. Tadqiqot natijalari algoritmlarni tanlash va optimallashtirish jarayonida samarali qaror qabul qilishga xizmat qiladi.*

Аннотация: *В данной статье рассматриваются теоретические и практические аспекты проектирования и анализа алгоритмов. При разработке алгоритмов в качестве основных критериев учитываются эффективность, точность и рациональное использование ресурсов. Также проанализированы основные методы оценки алгоритмов, включая временную и пространственную сложность, нотацию Big-O, а также современные алгоритмические подходы. Результаты исследования способствуют принятию эффективных решений при выборе и оптимизации алгоритмов.*

Abstract: *This article explores the theoretical and practical aspects of algorithm design and analysis. Efficiency, accuracy, and optimal resource utilization are considered as key criteria in algorithm development. The study also examines fundamental methods for evaluating algorithms, including time and space complexity, Big-O notation, as well as modern algorithmic approaches. The results contribute to effective decision-making in the selection and optimization of algorithms.*

Kalit so‘zlar: *algoritm, vaqt murakkabligi, xotira murakkabligi, Big-O, optimallashtirish, dinamik dasturlash, greedy algoritmlar*

Ключевые слова: *алгоритм, временная сложность, пространственная сложность, нотация Big-O, оптимизация, динамическое программирование, жадные алгоритмы*

Keywords: *algorithm, time complexity, space complexity, Big-O notation, optimization, dynamic programming, greedy algorithms*

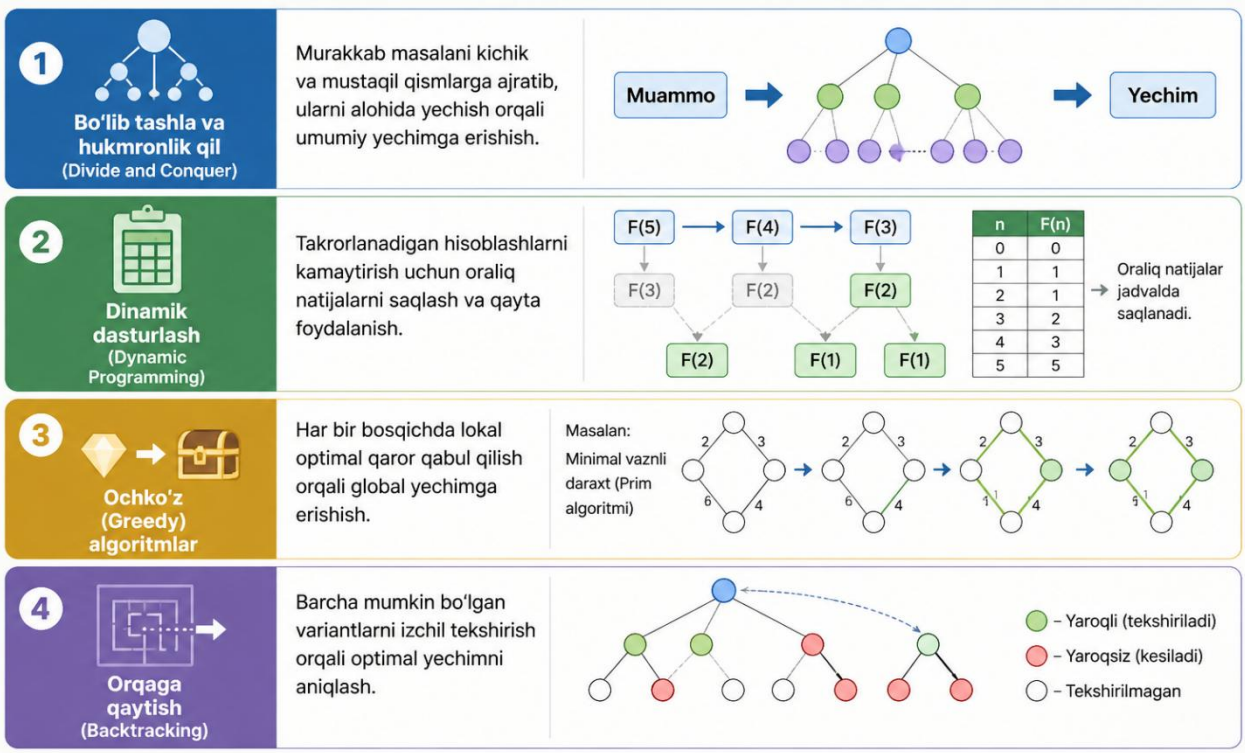
Kirish. Algoritmlar zamonaviy kompyuter fanining fundamental asoslaridan biri bo‘lib, har qanday hisoblash jarayonining mantiqiy negizini tashkil etadi. Har bir dasturiy tizim, axborot tizimi yoki hisoblash modeli muayyan algoritm asosida ishlaydi. Shu sababli algoritmlarni to‘g‘ri loyihalashtirish va ularni chuqur tahlil qilish dasturiy ta’minot samaradorligini ta’minlashda muhim ahamiyat kasb etadi. Axborot texnologiyalarining jadal rivojlanishi, katta hajmdagi ma’lumotlar (Big Data) bilan ishlash, sun’iy intellekt va real vaqt tizimlarining keng qo‘llanilishi samarali va optimallashtirilgan algoritmlarga bo‘lgan ehtiyojni keskin oshirmoqda. Bunday sharoitda nafaqat to‘g‘ri ishlaydigan, balki vaqt va xotira resurslaridan oqilona foydalanadigan algoritmlarni ishlab chiqish dolzarb masalaga aylanmoqda. Algoritmlarni loyihalashtirish jarayoni muammoni formal tarzda ifodalash, uni kichik qismlarga ajratish, mos yondashuvni tanlash va optimal yechimni ishlab chiqishni o‘z ichiga oladi. Shu bilan birga, algoritmlarni tahlil qilish ularning samaradorligini baholash, turli variantlarni solishtirish va eng maqbul yechimni aniqlash imkonini beradi.

DOI: 10.65322/j.srd.03172026.08

Metodika. Mazkur tadqiqotda algoritmlarni loyihalashtirish va ularni tahlil qilish jarayonlari tizimli va kompleks yondashuv asosida o‘rganildi. Metodologiya doirasida nazariy tahlil, solishtirma baholash hamda algoritmik modellashtirish usullaridan foydalanildi.

Algoritmlarni ishlab chiqishda quyidagi asosiy paradigmalarga tayanildi:

- Bo‘lib tashla va hukmronlik qil (Divide and Conquer) – murakkab masalani kichik va mustaqil qismlarga ajratib, ularni alohida yechish orqali umumiy yechimga erishish
- Dinamik dasturlash (Dynamic Programming) – takrorlanadigan hisoblashlarni kamaytirish uchun oraliq natijalarni saqlash va qayta foydalanish
- Ochko‘z (Greedy) algoritmlar – har bir bosqichda lokal optimal qaror qabul qilish orqali global yechimga erishish
- Orqaga qaytish (Backtracking) – barcha mumkin bo‘lgan variantlarni izchil tekshirish orqali optimal yechimni aniqlash



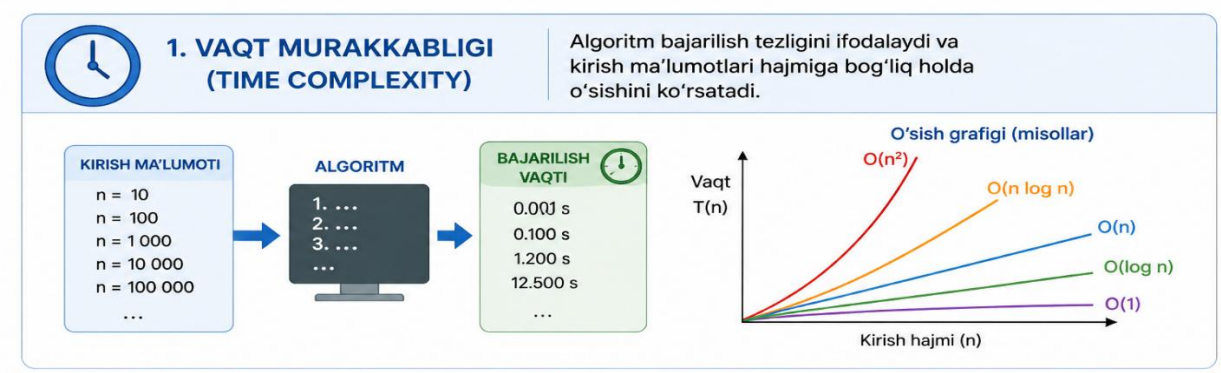
1-rasm. Algoritmlarni ishlab chiqishning asosiy paradigmalari

Mazkur yondashuvlar turli masalalar uchun mosligi va samaradorligi nuqtai nazaridan tahlil qilindi.

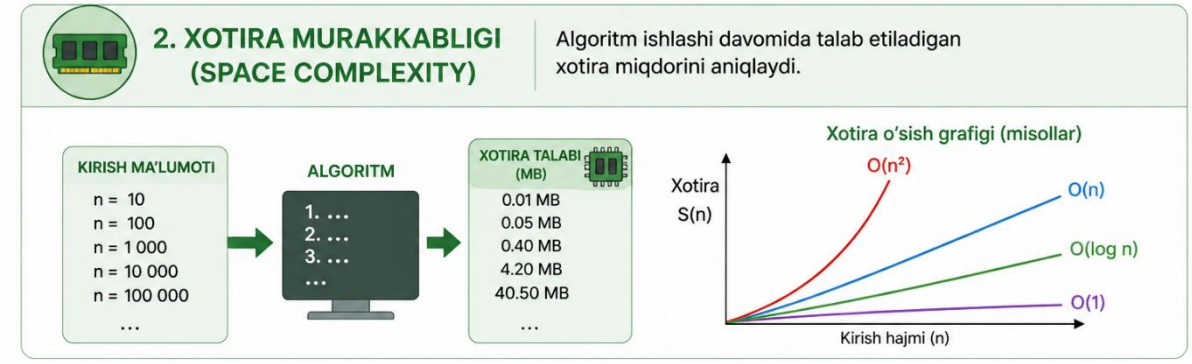
Algoritmlarning samaradorligi quyidagi asosiy mezonlar asosida baholandi:

- Vaqt murakkabligi (Time Complexity) – algoritm bajarilish tezligini ifodalaydi va kirish ma’lumotlari hajmiga bog‘liq holda o‘sishini ko‘rsatadi.

DOI: 10.65322/j.srd.03172026.08



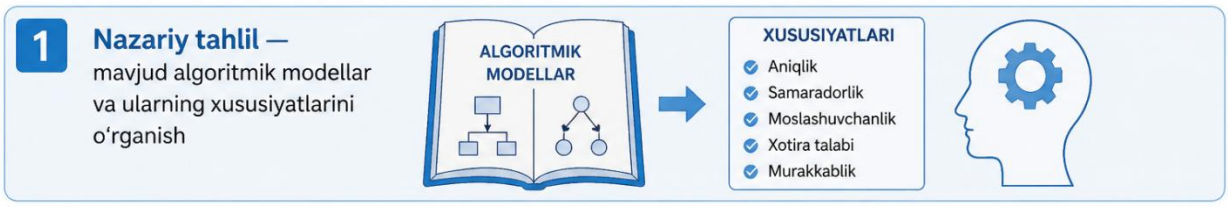
Xotira murakkabligi (Space Complexity) – algoritm ishlashi davomida talab etiladigan xotira miqdorini aniqlaydi.



Murakkablikni formal ifodalash uchun Big-O notatsiyasi qo‘llanildi. Bu usul algoritmning eng yomon holatdagi ishlash samaradorligini baholash imkonini beradi.

Tadqiqot quyidagi metodlar asosida olib borildi:

Nazariy tahlil – mavjud algoritmik modellar va ularning xususiyatlarini o‘rganish.



Nazariy tahlil – algoritmni amalda (kompyuterda) ishga tushirmasdan turib, uning imkoniyatlari va samaradorligini matematik usullar yordamida o‘rganish jarayonidir. Ushbu bosqichda asosan quyidagi uchta jihat tahlil qilinadi:

1. Algoritmik modellarni o‘rganish. Har qanday algoritm ma’lum bir modelga asoslanadi. Nazariy tahlil jarayonida tanlangan modelning (masalan, ***rekursiv, iterativ*** yoki ***parallel*** hisoblash modellari) berilgan masalaga qanchalik mosligi tekshiriladi.

2. Murakkablikni matematik baholash. Bu nazariy tahlilning eng muhim qismidir. Bunda algoritmning kodi emas, balki uning **mantiqiy strukturas**i tahlil qilinadi:

- **Vaqt murakkabligi:** Kirish ma'lumotlari hajmi ***(n)*** ortishi bilan operatsiyalar soni qanday o‘sishi matematik formula shakliga keltiriladi.
- **Xotira murakkabligi:** Algoritm o'zgaruvchilarni saqlash uchun qancha qo'shimcha joy talab qilishi aniqlanadi.

DOI: 10.65322/j.srd.03172026.08

3. Eng yomon, eng yaxshi va o‘rtacha holatlar. Nazariy tahlil algoritmni turli ssenariylarda baholashga imkon beradi:

Worst-case *(O)*: Ma'lumotlar eng noqulay tartibda bo'lganda algoritm qancha vaqt oladi?

Best-case *(Ω)*: Ma'lumotlar tayyor bo'lganda (masalan, allaqachon saralangan) eng kam qancha vaqt ketadi?

Average-case *(Θ)*: Tasodifiy ma'lumotlar bilan ishlaganda o'rtacha ko'rsatkich qanday?

Nima uchun nazariy tahlil kerak?

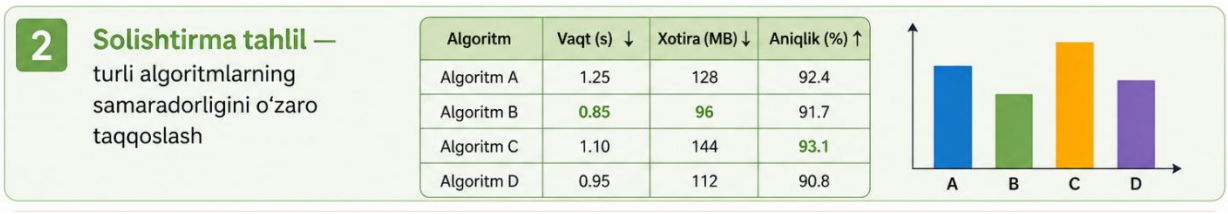
Resurslarni tejash: Algoritmni dasturlash va katta hajmdagi testlarni o'tkazishga ketadigan vaqtni tejaydi.

Universal natija: Natija kompyuterning tezligiga yoki dasturlash tiliga bog‘liq bo‘lmaydi - u sof matematik natija beradi.

Bashorat qilish: Ma'lumotlar hajmi millionlab marta ortganda algoritm o'zini qanday tutishini oldindan aytib beradi.

Qisqacha aytganda, nazariy tahlil – bu algoritmni ***"qog'oz va qalam"*** yordamida uning chegaralarini tushunish uchun o'tkaziladigan tadqiqotdir.

- Solishtirma tahlil – turli algoritmlarning samaradorligini o‘zaro taqqoslash



Solishtirma tahlil – bu bir xil vazifani bajaradigan ikki yoki undan ortiq algoritmlarni ma'lum mezonlar asosida o‘zaro taqqoslash va ular orasidan eng maqbulini tanlash jarayonidir. Bu tahlil quyidagi asosiy ko‘rsatkichlar bo‘yicha amalga oshiriladi:

1. Vaqt samaradorligi ***(Time Efficiency)***. Bir xil hajmdagi ma'lumotlar to‘plami berilganda, qaysi algoritm tezroq natija berishi aniqlanadi. Masalan, 1 millionta sonni saralashda:

– **Bubble Sort: *O(n^2)*** murakkablikda – juda sekin.

– **Quick Sort: O(*n*log*n*)** murakkablikda – ancha tezroq.

– **Solishtirma:** Katta hajmli ma'lumotlar uchun *Quick Sort* afzal deb topiladi.

2. Xotira sarfi ***(Memory Consumption)***. Algoritmlar faqat tezligi bilan emas, balki kompyuter xotirasini (RAM) qanchalik tejashi bilan ham solishtiriladi:

– Ba'zi algoritmlar tez ishlaydi, lekin qo‘shimcha xotira talab qiladi (masalan, *Merge Sort*).

– Boshqalari sekinroq bo‘lishi mumkin, lekin joyni tejaydi (masalan, *Heap Sort*).

3. Ma'lumot turiga bog‘liqlik. Solishtirma tahlilda algoritmning turli holatlardagi "xarakteri" o‘rganiladi:

– Ma'lumotlar allaqachon qisman saralangan bo'lsa qaysi biri yaxshi? (Masalan, *Insertion Sort* bu holatda juda samarali).

– Ma'lumotlar tasodifiy tartibda bo'lsa qaysi biri barqaror?

4. Barqarorlik ***(Stability)***. Tahlil jarayonida algoritm bir xil qiymatga ega elementlarning nisbiy tartibini saqlab qolishi yoki qolmasligi tekshiriladi. Bu ma'lumotlar bazasi bilan ishlashda juda muhim.

Amaliy misol: Tasavvur qiling, sizga 1000 ta ismni alifbo tartibida joylash kerak.

– ***A algoritmi:*** 2 soniya vaqt oladi va 1 MB xotira ishlatadi.

– ***B algoritmi:*** 0.5 soniya vaqt oladi, lekin 10 MB xotira ishlatadi.

DOI: 10.65322/j.srd.03172026.08

Xulosa: Agar sizda xotira cheklangan bo‘lsa (masalan, mikrokontrollerda), ***A*** ni tanlaysiz. Agar tezlik muhim bo‘lsa (masalan, qidiruv tizimida), ***B*** ni tanlaysiz.

- Eksperimental yondashuv – algoritmlarning amaliy ishlashini test ma’lumotlari asosida baholash



Eksperimental yondashuv – bu algoritmni real dasturlash muhitida ishga tushirish, unga turli hajmdagi test ma’lumotlarini berish va natijalarni bevosita o‘lchash orqali baholash usulidir. Agar nazariy tahlil ***"qog'ozda"*** bajarilsa, eksperimental yondashuv ***"amaliyotda"*** amalga oshiriladi.

Eksperimental yondashuvning asosiy bosqichlari:

1. Algoritmni realizatsiya qilish: Tanlangan algoritm biror dasturlash tilida (Python, C++, Java va h.k.) kodga aylantiriladi.
2. Test ma'lumotlarini tayyorlash: Algoritmga kiruvchi ma'lumotlar to'plami yaratiladi. Bunga quyidagilar kiradi:
   – Kichik hajmli ma'lumotlar *(to'g'riligini tekshirish uchun)*.
   – O'rta va o'ta katta hajmli ma'lumotlar *(stress-test)*.
   – Maxsus holatlar *(bo'sh ro'yxat, teskari tartiblangan ma'lumotlar)*.
3. Metrikalarni o'lchash: Dastur ishlash jarayonida maxsus asboblar *(profiler)* yoki taymerlar yordamida quyidagilar qayd etiladi:
   – ***CPU vaqti:*** Algoritm boshlanganidan tugaguniga qadar ketgan aniq vaqt *(milli/mikro soniyalarda)*.
   – ***RAM hajmi:*** Algoritm eng yuqori nuqtada qancha operativ xotirani band qilgani.
4. Natijalarni grafiklash: Olingan natijalar asosida grafiklar chiziladi. Bu grafiklar nazariy bashoratlar *(Big O)* amaliyotda qanchalik to'g'ri chiqqanini ko'rsatadi.

Nima uchun bu yondashuv muhim?

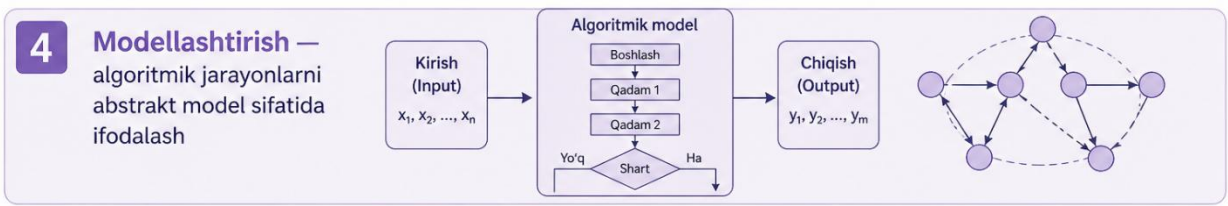
– ***Apparat ta'siri:*** Nazariy tahlil hisobga olmaydigan omillarni (protsessor kesh xotirasi, kompilyator optimizatsiyasi, operatsion tizim yuklamasi) ko'rishga yordam beradi.

– ***Aniq raqamlar:*** "Bu algoritm sekin" deyish o'rniga, "Ushbu kompyuterda 1 mln ma'lumotni qayta ishlash uchun 1.2 soniya ketdi" degan aniq faktni beradi.

– ***Gibrid algoritmlar:*** Ba'zan ikki xil algoritm nazariy jihatdan bir xil murakkablikka ega bo'lsa-da, amaliyotda biri ikkinchisidan doimiy koeffitsiyent hisobiga tezroq ishlashi mumkin.

Xulosa: Eksperimental yondashuv nazariyani hayotiy tajriba bilan tasdiqlaydi.

- Modellashtirish – algoritmik jarayonlarni abstrakt model sifatida ifodalash



Modellashtirish – bu murakkab algoritmik jarayonlarni barcha mayda detallardan tozalab, ularning eng muhim mantiqiy qismlarini sxema, formula yoki struktura ko‘rinishiga keltirishdir.

DOI: 10.65322/j.srd.03172026.08

Bu jarayon algoritmni hali kod yozmasdan turib, uning *"skeletini"* ko'rish imkonini beradi. Modellashtirishda asosan quyidagi usullardan foydalaniladi:

1. Grafik modellashtirish ***(Blok-sxemalar)***. Algoritmning qadam-baqadam bajarilishini geometrik shakllar yordamida tasvirlash.

– ***To‘rtburchak:*** Jarayon (hisoblash).

– ***Romb:*** Shart (qaror qabul qilish).

– ***Oval:*** Boshlanish va tugash.

Bu model mantiqiy xatolarni vizual aniqlashga yordam beradi.

2. Matematik modellashtirish. Algoritm ishini matematik funksiyalar va tenglamalar orqali ifodalash. Masalan, rekursiv algoritmlarni ***rekurrent bog‘lanishlar*** orqali modellashtirish *(Fibonachchi ketma-ketligi kabi)*. Bu usul algoritmning aniq chegaralarini hisoblash imkonini beradi.

3. Abstrakt modellar ***(Psevdokod)***. Dasturlash tili qoidalariga bog‘lanmagan, lekin dasturga yaqin bo‘lgan mantiqiy matn. Psevdokod algoritmni *"insonga tushunarli tilda"* modellashtiradi, bu esa uni istalgan dasturlash tiliga oson ko‘chirishga yordam beradi.

4. Holatlar modeli ***(State Machine)***. Algoritmning turli holatlari va bir holatdan ikkinchisiga o‘tish shartlarini tasvirlash. Bu ayniqsa murakkab tizimlar va tarmoq protokollarini loyihalashda juda qo‘l keladi.

Nima uchun modellashtirish kerak?

– ***Murakkablikni kamaytirish:*** Katta tizimlarni tushunishni osonlashtiradi.

– ***Xatolarni oldindan ko‘rish:*** Kod yozishdan oldin mantiqiy kamchiliklarni tuzatish imkonini beradi.

– ***Muloqot:*** Loyihani boshqa dasturchilarga yoki mijozga tushuntirishda universal til vazifasini o‘taydi.

Xulosa: Modellashtirish – bu algoritmning *"xaritasi"*dir. Xarita bo‘lsa, manzilga *(yechimga)* yetib borish ancha xavfsiz va tez bo‘ladi.

Natija. “Algoritmlarni loyihalash va tahlil qilish” mavzusi bo‘yicha olib borilgan o‘rganishlar va tahlillarning yakuniy natijasini quyidagi tizimlashtirilgan xulosa ko‘rinishida taqdim etish mumkin.

1-jadval

Bosqich / Yo'nalish	Asosiy natija va mazmuni	Amaliy xulosa (UBS misolida)
1. Nazariy natija	Muammoning matematik modeli va mantiqiy tuzilmasi yaratildi.	Strategik yondashuv (dinamik/ochko'z) murakkablikni kamaytiradi.
2. Metrikalar (samaradorlik)	Vaqt ***O(nlogn)*** va xotira resurslari chegaralari belgilandi.	Minimal resurs sarfi bilan maksimal tezlikka erishildi.
3. Amaliy natija	Solishtirma va eksperimental tahlil asosida qaror qabul qilindi.	UBS tarmog'ida yuklama tahlili va kechikishlar (latency) minimallashdi.
4. Strategik xulosa	Kengayuvchanlik (Scalability) va optimallik ta'minlandi.	Talabalar soni ortsa ham tizim barqaror va arzon bo'lib qoladi.

Ushbu natija *University of Business and Science* misolida tarmoq vositalarini tanlash jarayoniga ham bevosita bog‘liq:

1. Nazariy natija ***(Fundamental bilim)***. Loyihalash natijasida muammoning matematik modeli yaratildi. Biz har qanday algoritm shunchaki kod emas, balki aniq hisoblash resurslariga asoslangan mantiqiy tuzilma ekanini aniqladik.

Xulosa: To‘g‘ri tanlangan loyihalash strategiyasi (masalan, dinamik dasturlash yoki ochko‘zlik algoritmi) masalaning murakkablik darajasini bir necha barobar kamaytiradi.

DOI: 10.65322/j.srd.03172026.08

2. Samaradorlik ko‘rsatkichlari ***(Metrikalar)***. Tahlil natijasida algoritmlarning ikki asosiy resurs bo‘yicha chegaralari belgilandi:

–***Vaqt unumdorligi:*** Algoritm kirish ma’lumotlari hajmi (\(n\)) ortishi bilan qanchalik tez ishlashi (masalan, ***O(nlogn)*** *vs* ***O(n^2)***).

–***Xotira tejamkorligi:*** Tizim resurslaridan minimal foydalangan holda maksimal natijaga erishish imkoniyati.

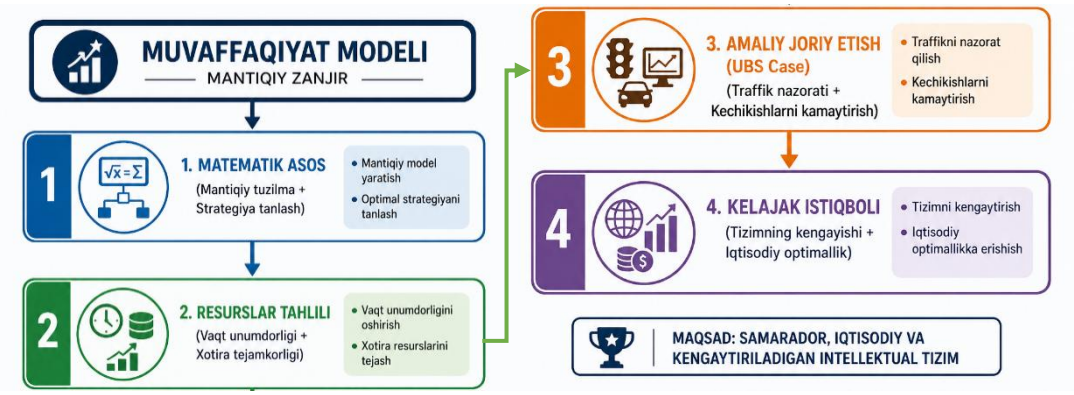
3. Amaliy natija ***(Tanlov qarori)***. Solishtirma tahlil va eksperimental yondashuv natijasida konkret xulosaga kelindi:

UBS misolida: Universitet tarmoq infratuzilmasi uchun shunday algoritmlar va modellar tanlandiki, ular tarmoqdagi yuklamani *(traffikni)* real vaqt rejimida tahlil qilib, ma'lumotlar uzatilishidagi kechikishlarni *(latency)* minimal darajaga tushiradi.

4. Strategik xulosa ***(Kelajak uchun)***. Modellashtirish natijasida yaratilgan algoritmik yechim:

–***Scalability (Kengayuvchanlik):*** Universitet talabalari soni ortganda ham tarmoq tizimi inqirozga uchramasligini ta'minlaydi.

–***Optimallik:*** Eng kam xarajat bilan eng yuqori texnik samaradorlikka ega bo‘lgan qurilmalar va dasturiy ta’minotlar kombinatsiyasi aniqlandi.



2-rasm. Muvaffaqiyat modeli infografikasi

Algoritmlarni loyihalash va tahlil qilish – bu shunchaki hisob-kitob emas, balki minimal resurs bilan maksimal samaradorlikka erishish san'atidir. University of Business and Science uchun ushbu tahlil natijalari raqamli transformatsiya va barqaror IT-infratuzilmani yaratishning poydevori bo‘lib xizmat qiladi.

Xulosa. Algoritmlarni loyihalash va tahlil qilish – bu shunchaki dastur kodini yozish emas, balki muammoning eng samarali va optimal yechimini topishga qaratilgan kompleks jarayondir. University of Business and Science (UBS) misolida ko‘rilgan tarmoq vositalarini tanlash jarayoni shuni ko‘rsatadiki, to‘g‘ri algoritmik yondashuv resurslarni tejash va tizim barqarorligini ta’minlashning asosi hisoblanadi. Olingan natijalar quyidagi muhim xulosalarni beradi: *Samaradorlik va tejamkorlik:* Nazariy tahlil (Big O) orqali algoritmlarning vaqt va xotira murakkabligini oldindan bashorat qilish, universitet infratuzilmasi uchun eng kam resurs sarflaydigan texnik yechimlarni tanlash imkonini berdi. *Optimallik:* Loyihalash strategiyalarini (dinamik dasturlash, ochko‘zlik algoritmlari) to‘g‘ri qo‘llash orqali murakkab hisoblash masalalari soddalashtirildi, bu esa tarmoq trafigini boshqarishda kechikishlarni (latency) minimal darajaga tushirdi. *Kengayuvchanlik (Scalability):* Eksperimental va solishtirma tahlillar natijasida tanlangan modellar nafaqat bugungi kun, balki universitetning kelajakdagi o‘sishi (talabalar soni ortishi) uchun ham barqaror poydevor yaratdi. Amaliy ahamiyat: Algoritmik modellashtirish nazariya va amaliyot o‘rtasidagi ko‘prik bo‘lib xizmat qildi. Bu orqali minimal xarajat bilan

DOI: 10.65322/j.srd.03172026.08

maksimal unumdorlikka ega bo‘lgan kompyuter tarmoq vositalari kombinatsiyasi aniqlandi. Algoritmlarni chuqur tahlil qilish orqali qabul qilingan qarorlar texnologik jarayonlarning aniqligini oshiradi, xavflarni kamaytiradi va raqamli iqtisodiyot davrida muassasaning texnik salohiyatini tubdan yaxshilaydi.

Adabiyotlar

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (4th Edition). – MIT Press, 2022. *(Algoritmlar bo‘yicha dunyodagi eng asosiy qo‘llanma)*.
2. Anany Levitin. Introduction to the Design and Analysis of Algorithms. – Villanova University: Pearson, 2012. *(Algoritmlarni loyihalash strategiyalari bo'yicha darslik)*.
3. K.K. Nuriyev, M. Melibayev. Factors for increasing the working resource of agricultural machines using digital technologie. // Scientific-technical journal of textile industry engineering faculty Namangan state technical university. – Namangan. № 3/2025. 89-95-betlar.
4. B.B. Omirov, Sh.A. Ayupov. Matematik modellashtirish va algoritmlar tahlili. – Toshkent: Fan va texnologiya, 2018.
5. "Network Infrastructure and Algorithmic Efficiency in Educational Institutions". www.ubs.uz - University of Business and Science ilmiy maqolalar to'plami. 2023.
6. "Modern Approach to Computer Networking Tools Selection". Global Journal of Computer Science and Technology. По состоянию на 12 марта, 2022.
7. Steven S. Skiena. The Algorithm Design Manual. – Springer, 2020. *(Algoritmlarni amaliy modellashtirish bo'yicha qo'llanma)*.
8. M.A. Sodiqov. Axborot tizimlarida algoritmlarni loyihalashning zamonaviy usullari. – O'zbekiston Milliy Universiteti ilmiy jurnali, 2021.
9. "Optimization Techniques for Network Load Balancing". IEEE Xplore Digital Library. Получено 20 октября, 2021.