



# TOSHKENT AMALIY FANLAR UNIVERSITETI

## Ma'lumotlar tuzilmasi va algoritmlar fani

“Kompyuter injiniring” kafedrası  
Katta o'qituvchi Kendjayeva Dildora  
Xudayberganovna

# Birlashtirib saralash algoritmlari





# MA'RUZA REJASI



**Merge sort algoritmi**

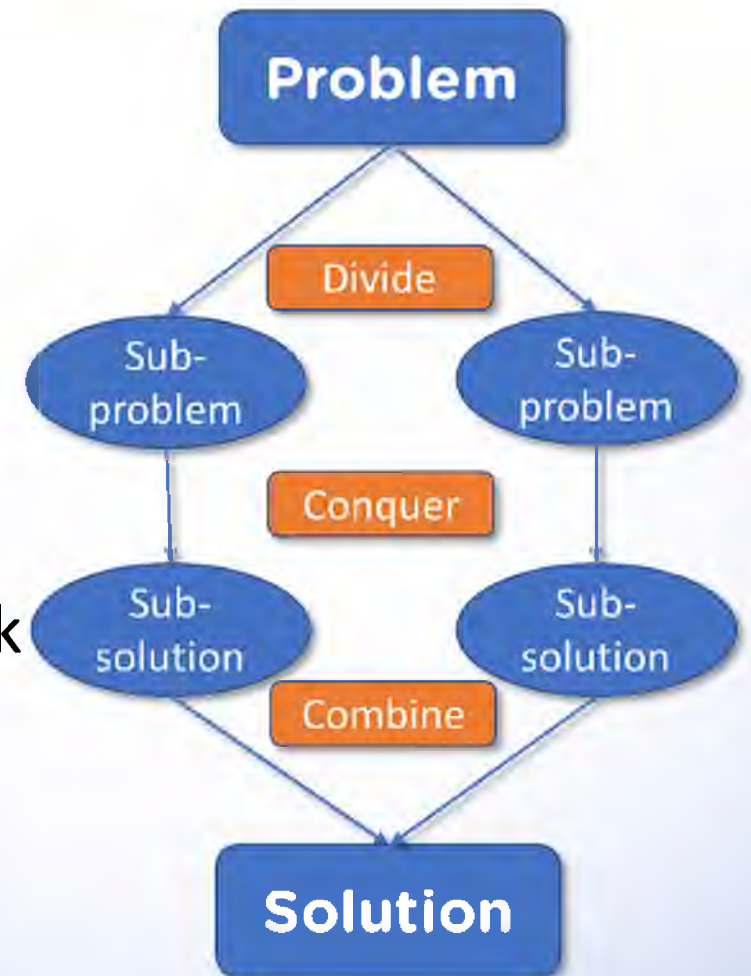
**“Bo’lib tashla va hukmronlik qil” strategiyasi**

**Birlashtirib saralash algoritmi**

**Birlashtirib saralash algortimini baholash**

# Merge sort algoritmi.

- **Birlashtirib saralash (Merge sort)** - tartiblashning tezkor bajariladigan algoritmlaridan biri. Ushbu tartiblash “**bo‘lib tashla va hukmronlik qil**” prinsipining yaxshi namunasidir.
- Birinchidan, vazifa bir nechta kichik topshiriqlarga bo‘linadi.
- Keyin ushbu vazifalar rekursiv chaqiruv yordamida yoki to‘g‘ridan-to‘g‘ri ularning hajmi yetarlicha kichik bo‘lsa hal qilinadi.
- Nihoyat, ularning yechimlari birlashtirilib, asl muammoning echimi olinadi.



# Algoritmning bajarilishi

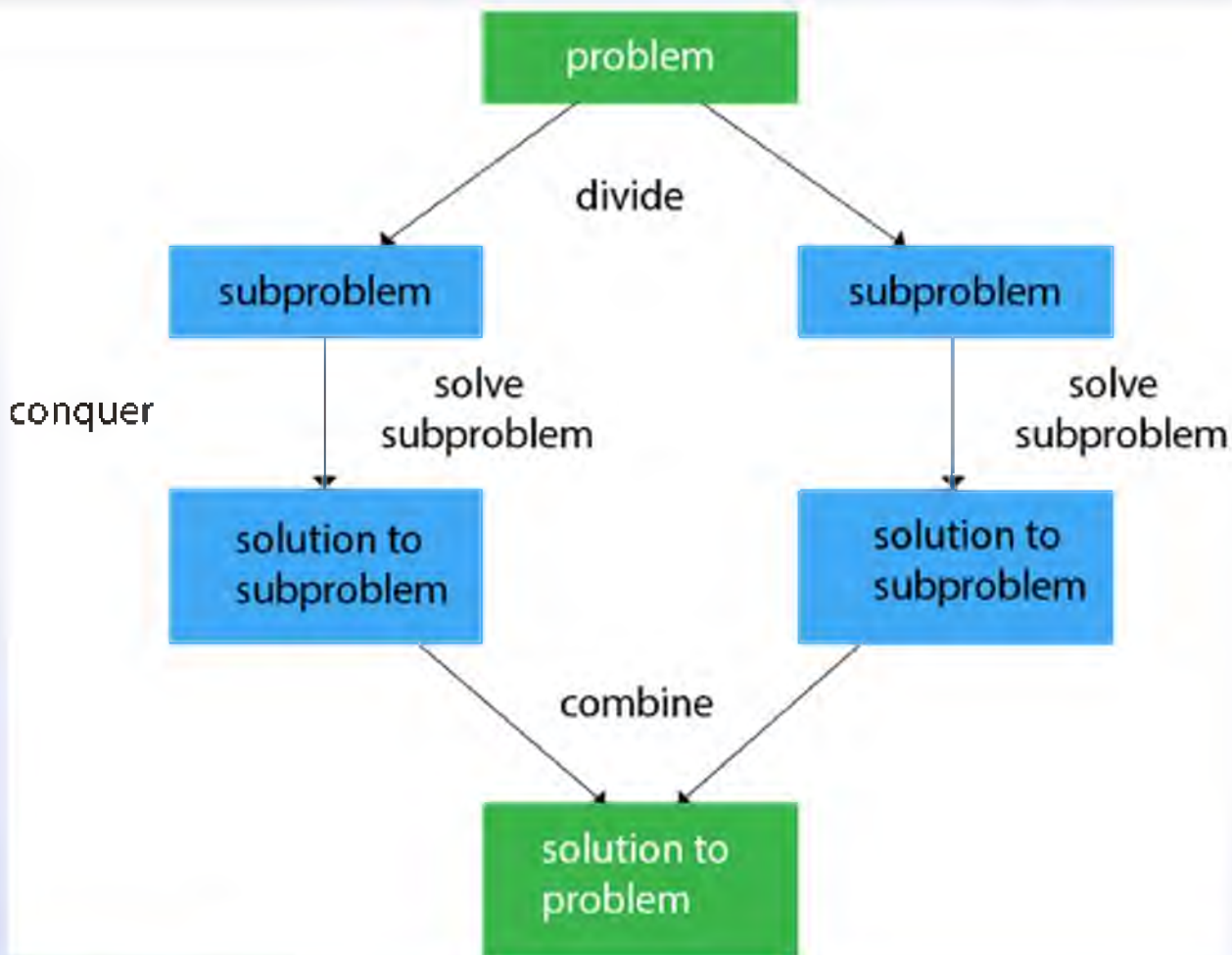
Saralash muammosini hal qilish uchun **uch bosqich** quyidagicha bo'ladi:

1. Saralanadigan massiv taxminan bir xil o'lchamdagi **ikki qismga bo'linadi**;
2. Olingan qismlarning har biri **alohida saralanadi** (masalan, xuddi shu algoritm bo'yicha saralanadi);
3. Yarim kattalikdagi ikkita saralangan **massivlar birlashtiriladi**.

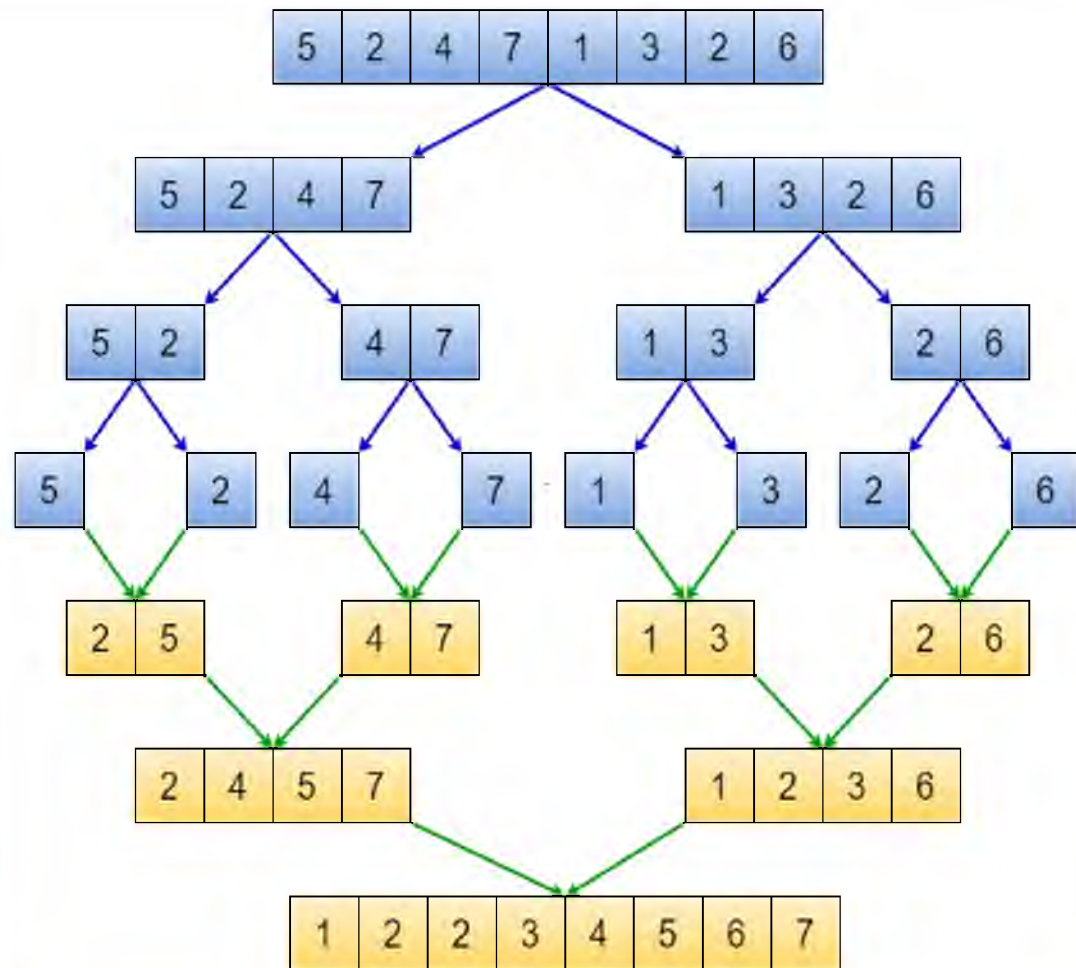
Bu eng mashhur saralash algoritmlaridan biri bo'lib, rekursiv algoritmlarni yaratishda ishonchni rivojlantirishning ajoyib usuli hisoblanadi.



# “Bo‘lib tashla va hukmronlik qil” strategiyasi



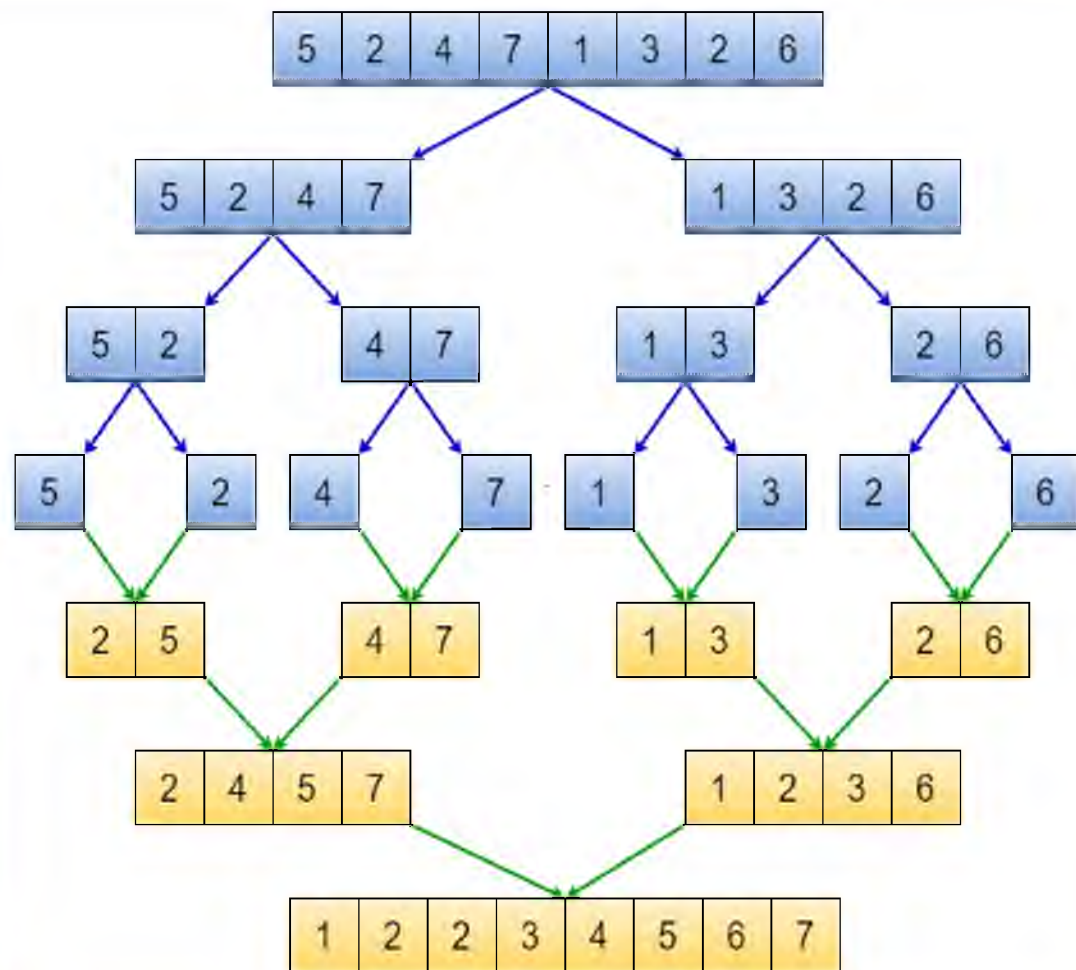
- “Bo‘lib tashla va hukmronlik qil” (divide & conquer) strategiyasi yordamida muammoni qisman jarayonlarga ajratamiz.
- Har bir kichik topshiriq uchun yechimga ega bo‘lsak, pastki vazifalarni yechish uchun pastki vazifalardan olingan natijalarni "birlashtiramiz".



**“Bo'lib tashlash”.** Agar  $q$  qiymati  $p$  va  $r$  orasida bo'lsa, biz  $A[p..r]$  massivni ikkita  $A[p..q]$  va  $A[q + 1, r]$  kichik massivlarga bo'lishimiz mumkin.

**“Hukmronlik qilish”.** —Hukmronlik qilish” bosqichida biz ikkala  $A[p..q]$  va  $A[q + 1, r]$  kichik massivlarni saralashga harakat qilamiz. Agar hali ham boshlang'ich darajaga yetib bormagan bo'lsak, yana ikkala quyi qismni ajratib, ularni saralashga harakat qilamiz.

# Birlashtirish bosqichi



Birlashtirish bosqichi asosiy pog'onaga yetib borganida va biz  $A[p..r]$  massivi uchun ikkita tartiblangan  $A[p..q]$  va  $A[q + 1, r]$  kichik massivlarni olsak, natijalarni  $A[p..r]$  massiviga birlashtiramiz.

Bu ikkita tartiblangan  $A[p..q]$  va  $A[q + 1, r]$  massivlarning birlashmasidir



# Birlashtirib saralash algoritmi.

**MergeSort funksiyasi** massivni ketma-ket ikki qismga ajratadi, biz 1-darajali ichki massivda MergeSort- ga o'tishga harakat qiladigan bosqichga yetguncha ya'ni  $p == r$  bo'lguncha.

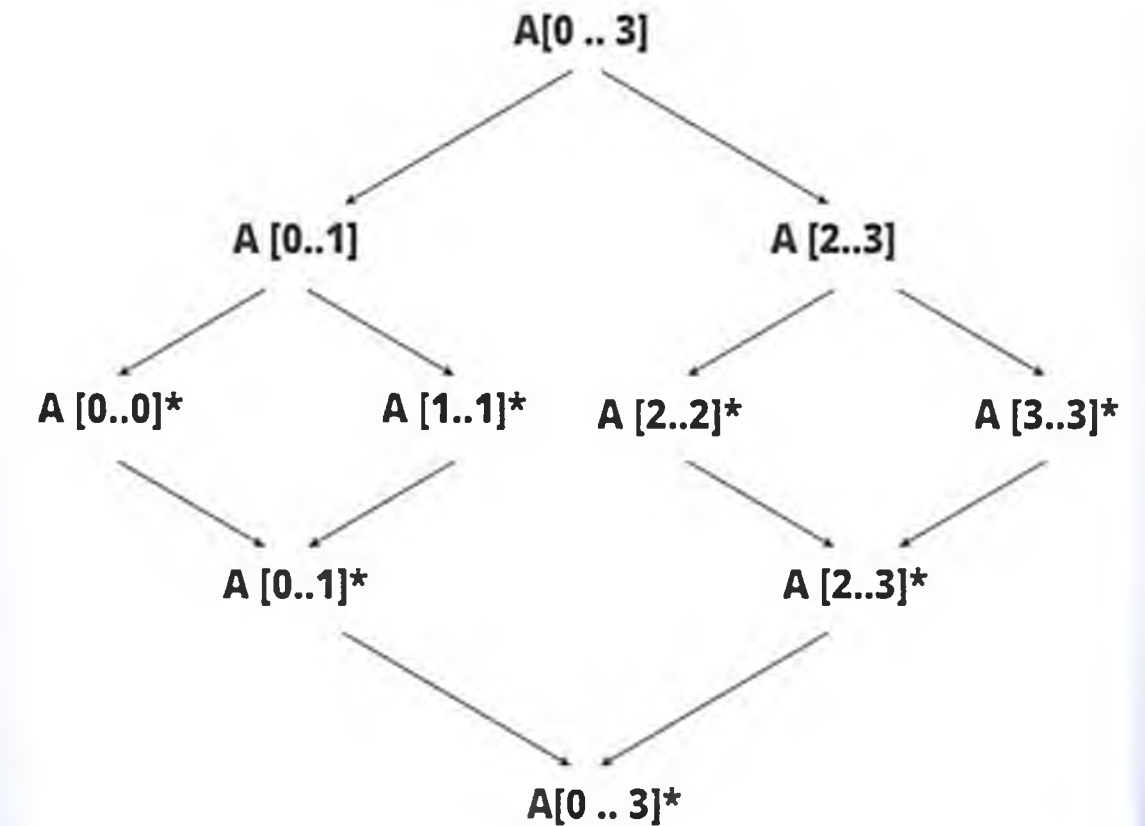
Shundan so'ng, birlashtirish funksiyasi ishga tushadi, bu tartiblangan massivlarni butun massiv birlashguncha katta massivlarga birlashtiradi.

Butun massivni saralash uchun MergeSort ( $A, 0, \text{length}(A) - 1$ ) ga murojaat qilishimiz kerak.

1. MergeSort( $A, p, r$ )
2. If  $p > r$
3. return;
4.  $q = (p+r)/2$ ;
5. mergeSort( $A, p, q$ )
6. mergeSort( $A, q+1, r$ )
7. merge( $A, p, q, r$ )

Rasmda ko'rsatilgandek, birlashtirib saralash algoritmi 1 elementli massivning asosiy holatiga kelgunimizcha massivni rekursiv ravishda ikkiga bo'ladi.

So'ngra birlashtirish funksiyasi saralangan ichki massivlarni tanlaydi va butun qatorni asta-sekin saralash uchun ularni birlashtiradi.



- Algoritmning eng muhim qismi bu "birlashtirish" bosqichidir. Birlashish bosqichi - ikkita katta ro'yxat (massiv) yaratish uchun ikkita tartiblangan ro'yxatni (massivlarni) birlashtirish bo'yicha oddiy muammoning yechimi.
- Ikkinchi massivda boshqa elementlar qolmaganligi va ikkala massiv ham ishga tushirilganda saralanganligini bilganimiz uchun qolgan massivlarni to'g'ridan-to'g'ri birinchi massivdan nusxalashimiz mumkin.





# Birlashtirib saralash algortimini baholash.

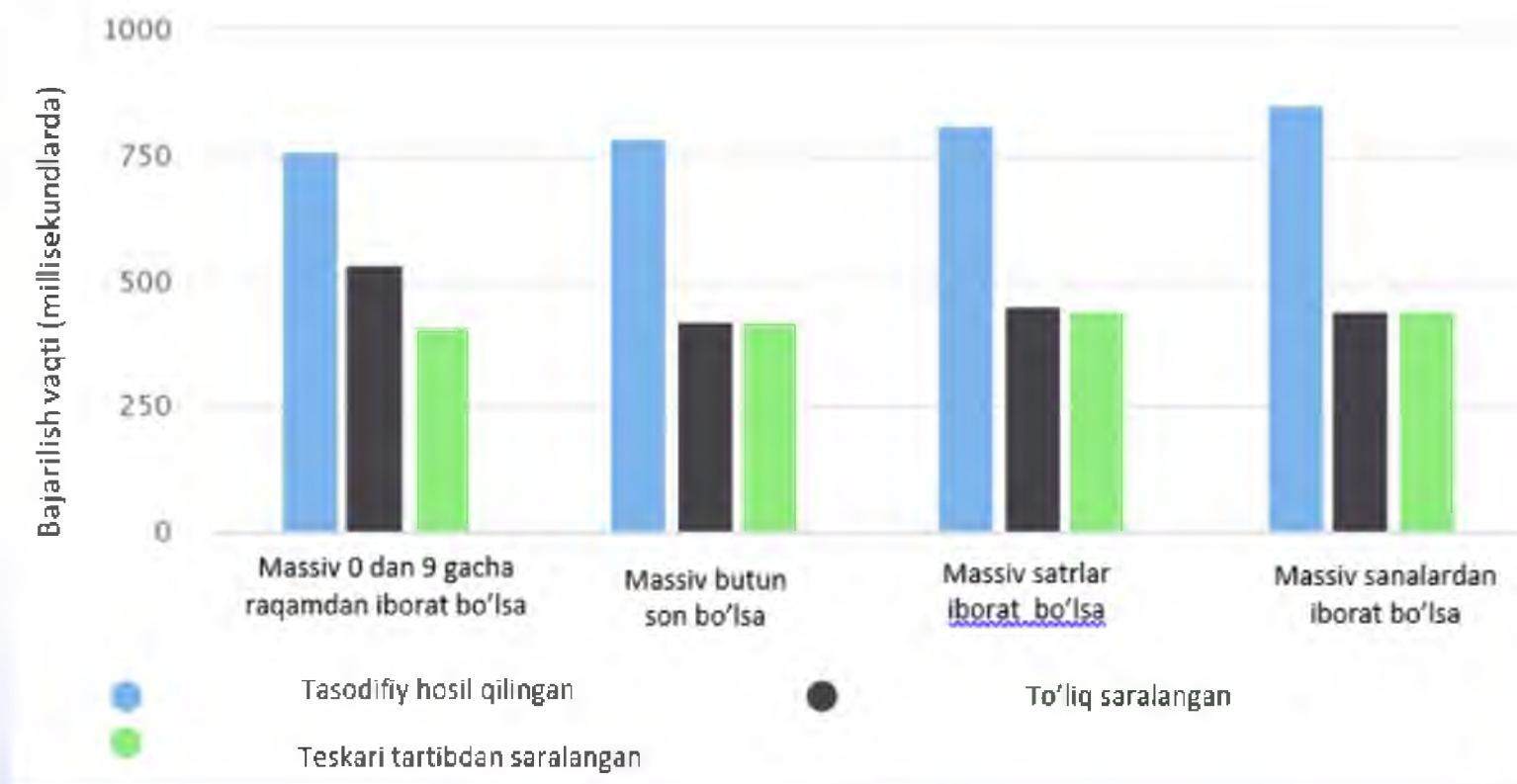
Algoritmnining murakkabligini taxmin qilaylik. Har qanday **rekursiv funksiya chaqiruvi daraxtga** o'xshaydi (Izoh: "Daraxtlar" haqida keyingi ma'ruzalarda to'xtalib o'tiladi). Bunday daraxtni **rekursion daraxt** deb ham atashadi.

Daraxtning **har bir darajasi** bir yoki bir nechta **funksiya chaqiruvlarini** aks ettiradi. Shoxlari bo'lmagan daraxt tugunlari rekursiyani tugatadigan funksiya chaqiruvlarini anglatadi.

**Birlashtirish tartibida** daraxtning balandligi  **$\log_2 n$**  ga teng, chunki har bir qadamda boshlang'ich massiv  **$n/2$**  uzunlikdagi ikkita ichki massivga bo'linadi. Ajratishdan so'ng, birlashtirish operatsiyasi amalga oshiriladi.

Birlashtirish jarayoni  **$n$**  taqqoslashni, navbati bilan  **$\log n$**  marta, ya'ni daraxtning har bir darajasi uchun 1 marta takrorlashni talab qiladi. Keyin algortim asimptotikasi  **$O(n \log n)$**  bo'ladi.

# Merge Sort algoritmining turli xil tiplar uchun ishlash vaqti (elementlar soni 50000 ta)



---

**Do you have  
any questions?**

---

