Розробка та аналіз алгоритмів

3. Метод декомпозиції

- Сортування злиттям
- Підрахунок інверсій
- Множення матриць

Метод Карацуби множення цілих чисел

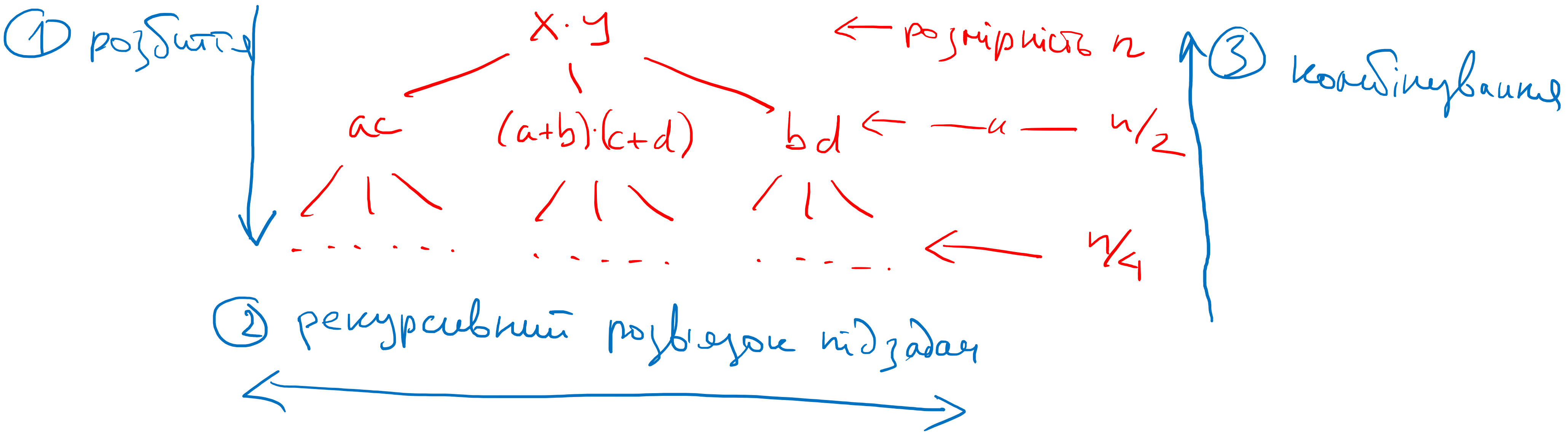
Вхід: 2 n-розрядних числа X та Y X: $\boxed{a|b}$ Y: $\boxed{c|d}$

Вихід: $Z = X \cdot Y$

$X = 10^{n/2} a + b \qquad Y = 10^{n/2} c + d$

$$Z = X \cdot Y = 10^{n} \underline{ac} + 10^{n/2}(ad+bc) + \underline{bd}$$

$$ad + bc = \underbrace{(a+b)}\cdot\underbrace{(c+d)} - ac - bd$$



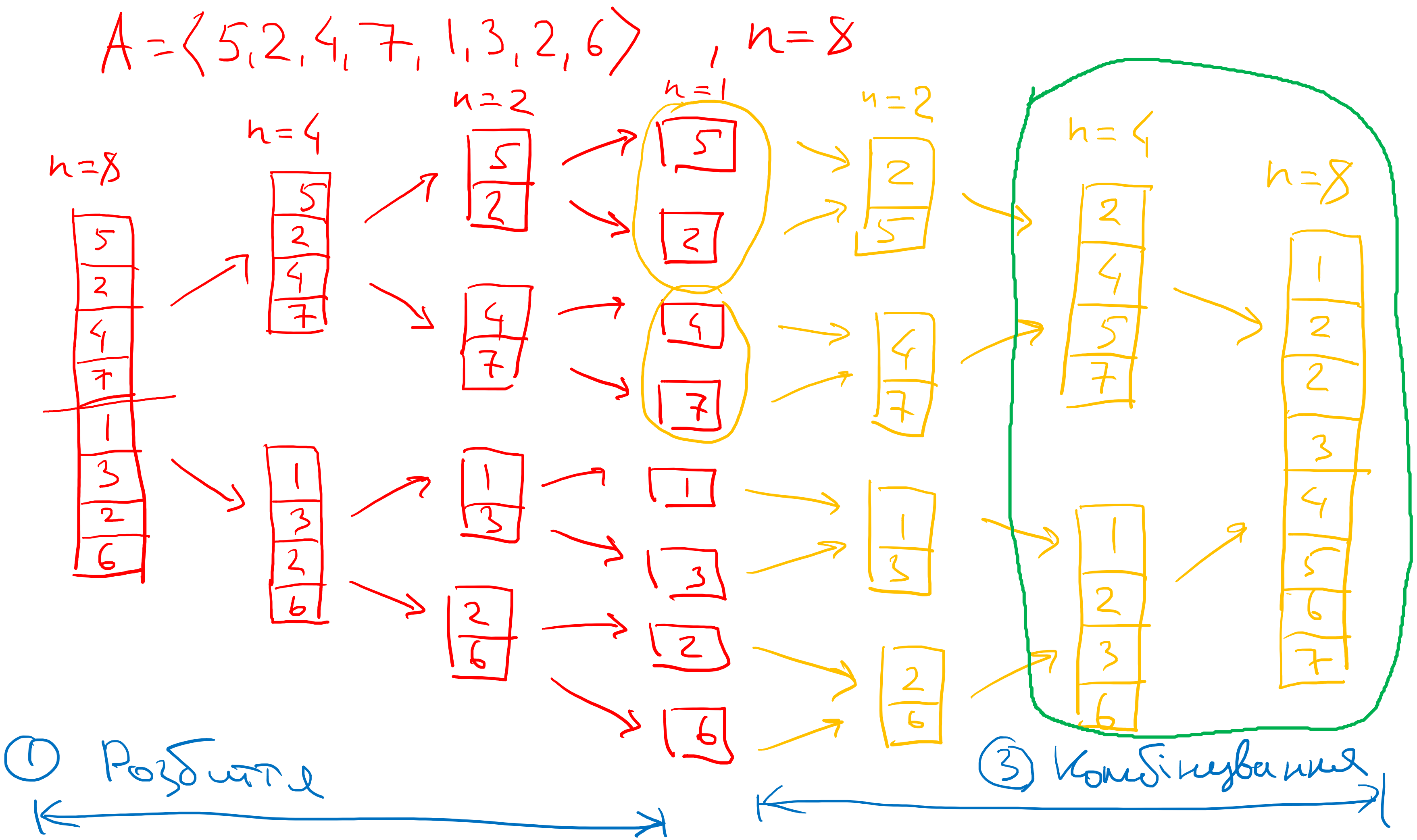
Сортування методом злиттям (Merge Sort)

Вхід: масив $A = \langle a_1, \dots, a_n \rangle$

Вихід: $A = \langle a_1', \dots, a_n' \rangle$, де $a_1' \leq a_2' \leq \dots \leq a_n'$

- **Розділення**: послідовність для сортування, яка складається з *n* елементів, розбивається на дві менші послідовності, кожна з яких містить *n*/2 елементів
- **Рекурсивний розв'язок**: сортування обох створених послідовностей методом злиття, якщо їх розмір перевищує 1
- **Комбінування**: злиття двох відсортованих послідовностей для отримання кінцевого результату

Приклад сортування злиттям



Алгоритм методу сортування злиттям

Вхід: A – масив, p, r – індекси ⇒ A[p...r] – підмасив для сортування

Вихід: A[p...r] – відсортований підмасив

MergeSort (A, p, r)

1. if $p<r$ then:
2. $q = \lfloor (p+r)/2 \rfloor$
3. MergeSort (A, p, q) ← сортування лівої частини
4. MergeSort (A, q+1, r) ← сортування правої частини
5. Merge (A, p, q, r) ← злиття двох частин

MergeSort (A, 1, n) ← перший виклик

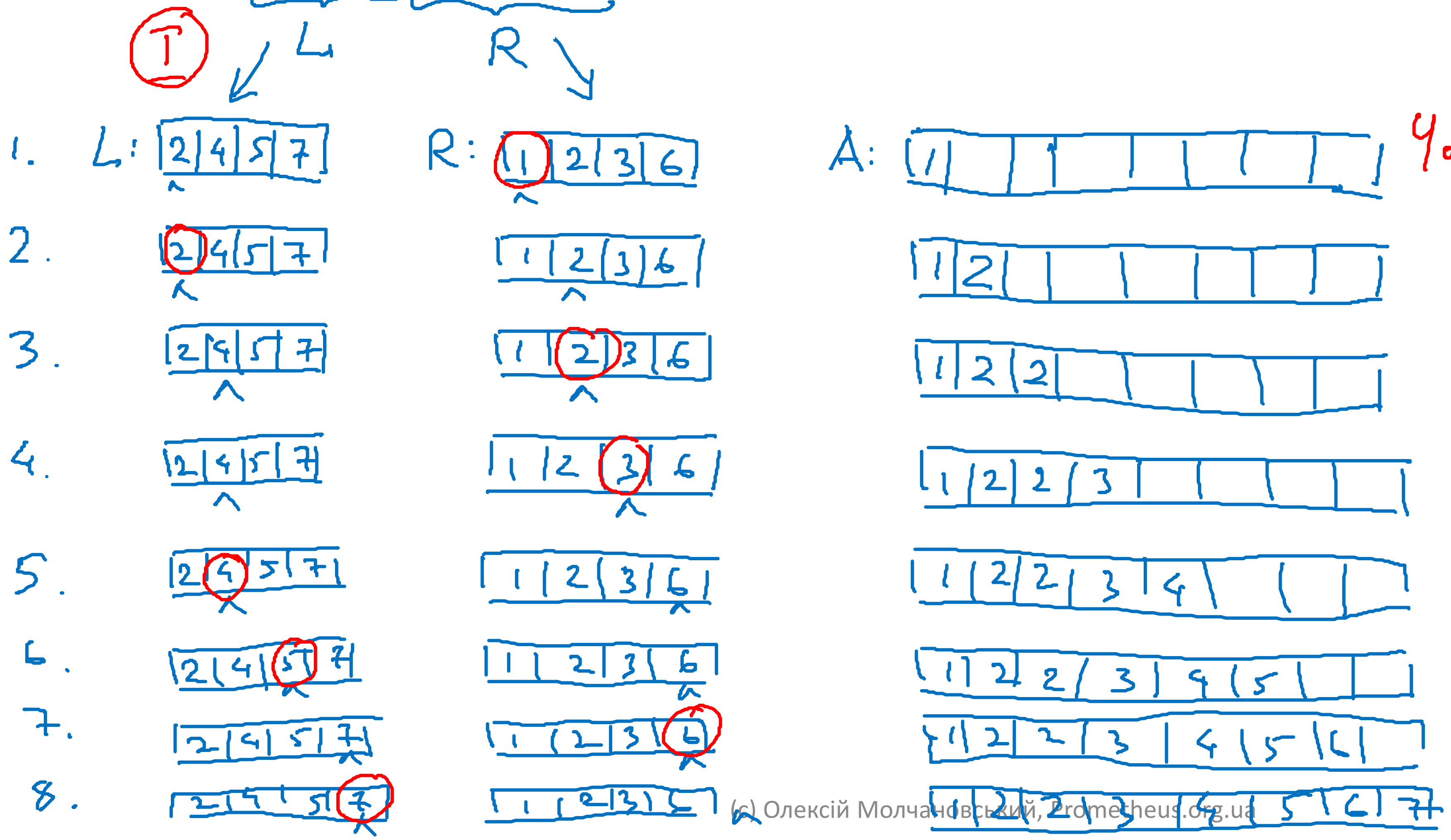
Приклад:

p=1, r=6	p=1 r=7
q=3	q=4
A[1..3]	A[1..4]
A[4...6]	A[5..7]

Приклад роботи процедури злиття

Вхід: A = | 2 | 4 | 5 | 7 | 1 | 2 | 3 | 6 |, p = 1, q = 4, r = 8

$n = 8$



Час: $\Theta(n)$

Процедура злиття

Вхід: масив A, індекси початку, середини та кінця поточного розбиття: p, q, r

Вихід: масив A із відсортованою частиною між p та r

Merge(A, p, q, r)

1. $n1 = q - p + 1$
2. $n2 = r - q$
3. Створити масиви $L[1 \dots n1+1]$ та $R[1 \dots n2+1]$ $\Theta(1)$
4. for $i=1$ to $n1$: $L[i] = A[p+i-1]$ $\Theta(n1)$
5. for $j=1$ to $n2$: $R[j] = A[q+j]$ $\Theta(n2)$
6. $L[n1+1] = \infty$, $R[n2+1] = \infty$
7. $i = 1$
8. $j = 1$ $\Theta(1)$
9. for $k=p$ to r: $\Theta(n)$
10. if $L[i] \le R[j]$ then:
11. $A[k] = L[i]$
12. $i = i+1$
13. else:
14. $A[k] = R[j]$
15. $j = j+1$ $\Theta(1)$

Час роботи Merge: $\Theta(n_1) + \Theta(n_2) + \underline{\Theta(n)} + c \cdot \Theta(1) = \Theta(n)$

Рекурентне рівняння

$T(n)$ – час розв'язку задачі розміром n

1. $D(n)$ – час для розбиття задачі на підзадачі
2. задача розбивається на a підзадач, об'єм кожної дорівнює $1/b$ від розміру початкової
3. $C(n)$ – час на комбінування розв'язків підзадач

При $n \le c$ задача розв'язується за сталий час: $T(n) = \Theta(1)$

$$\boxed{T(n) = \begin{cases} \Theta(1), & \text{при } n \le c \\ D(n) + C(n) + a \cdot T(n/b), & \text{при } n > c \end{cases}}$$

Merge Sort

1. $D(n) = \Theta(1)$
2. $a = 2, \ b = 2$
3. $C(n) = \Theta(n)$

$c = 1$

$$T(n) = \begin{cases} \Theta(1), & \text{при } n = 1 \\ \underbrace{\Theta(1) + \Theta(n)} + 2 \cdot T(n/2) = \end{cases}$$

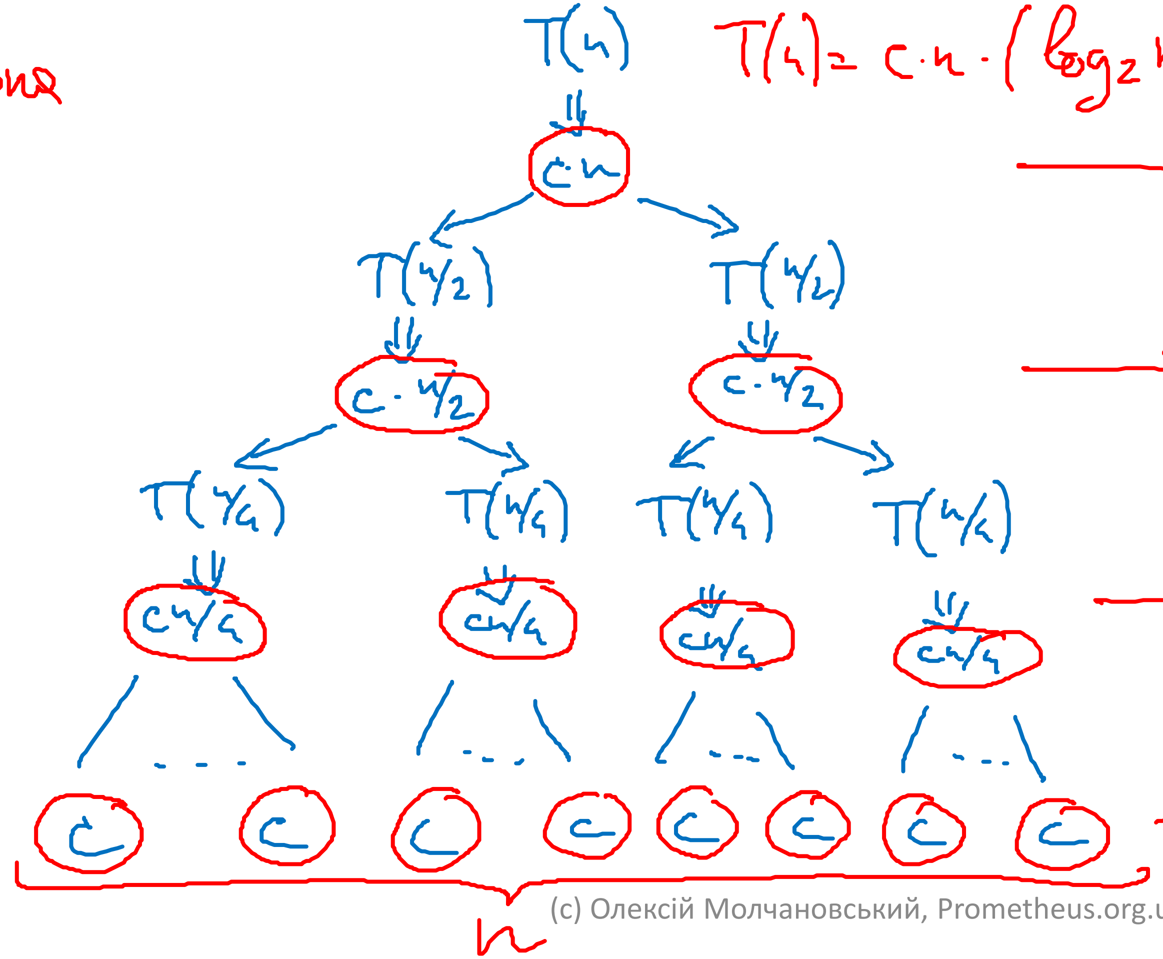
$$= \Theta(n) + 2 \cdot T(n/2)$$

Рекурентне рівняння для методу злиття

$\Theta(n) = c\cdot n$, c – const.

$$T(n) = \begin{cases} c, & \text{при } n=1 \\ cn + 2\cdot T(n/2), & n>1 \end{cases}$$

$T(n) = c\cdot n\cdot(\log_2 n + 1) = cn\log_2 n + cn = \Theta(n\lg n) + \Theta(n) = \Theta(n\cdot\lg n)$

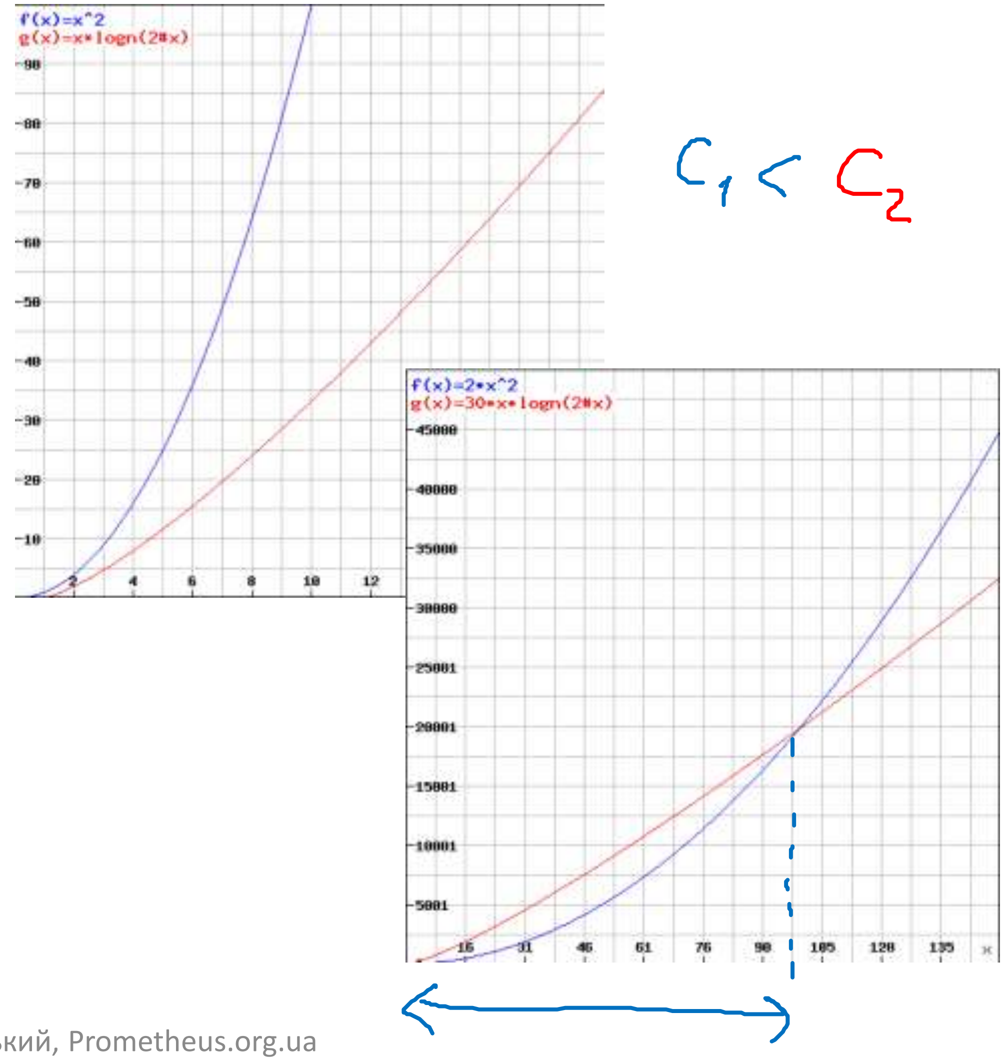


Кількість вузлів	# рівня	
1	0	$\to c\cdot n$
2	1	$\to 2\cdot\left(c\cdot\frac{n}{2}\right) = cn$
4	2	$\to 4\cdot\left(c\cdot\frac{n}{4}\right) = cn$
2^i ⋮	i ⋮	
		$\to n\cdot c = cn$

$2^i = n \Rightarrow i = \boxed{\log_2 n}$

Порівняння сортування включенням та злиттям

	InsertionSort	MergeSort
Час роботи	$\Theta(n^2) = c_1 \cdot n^2$	$\Theta(n \lg n) = c_2 \cdot n \lg n$
Пам'ять	не використовує додаткову пам'ять	використовує додаткову пам'ять $\Theta(n)$



$c_1 < c_2$

Задача порівняння вподобань

Netflix

Аліса		Богдан
1	Пірати Карибського моря	5
2	Володар перстнів	2
3	Матриця	4
4	Гравітація	3
5	Зоряні війни	1

Наскільки близькими (схожими) є вподобання Аліси та Богдана?

Задача підрахунку інверсій

$A = [1, \ldots, n]$ – масив з n чисел

Якщо $i < j$ та $A[i] > A[j]$, що пара (i, j) називається інверсією в масиві A.

Задача: Скільки інверсій в заданому масиві A?

	1	2	3	4	5
$A =$	2	3	8	6	1

Інверсії в A: (1,5), (2,5), (3,4), (3,5), (6,1) $K = 5$

Крайні випадки:

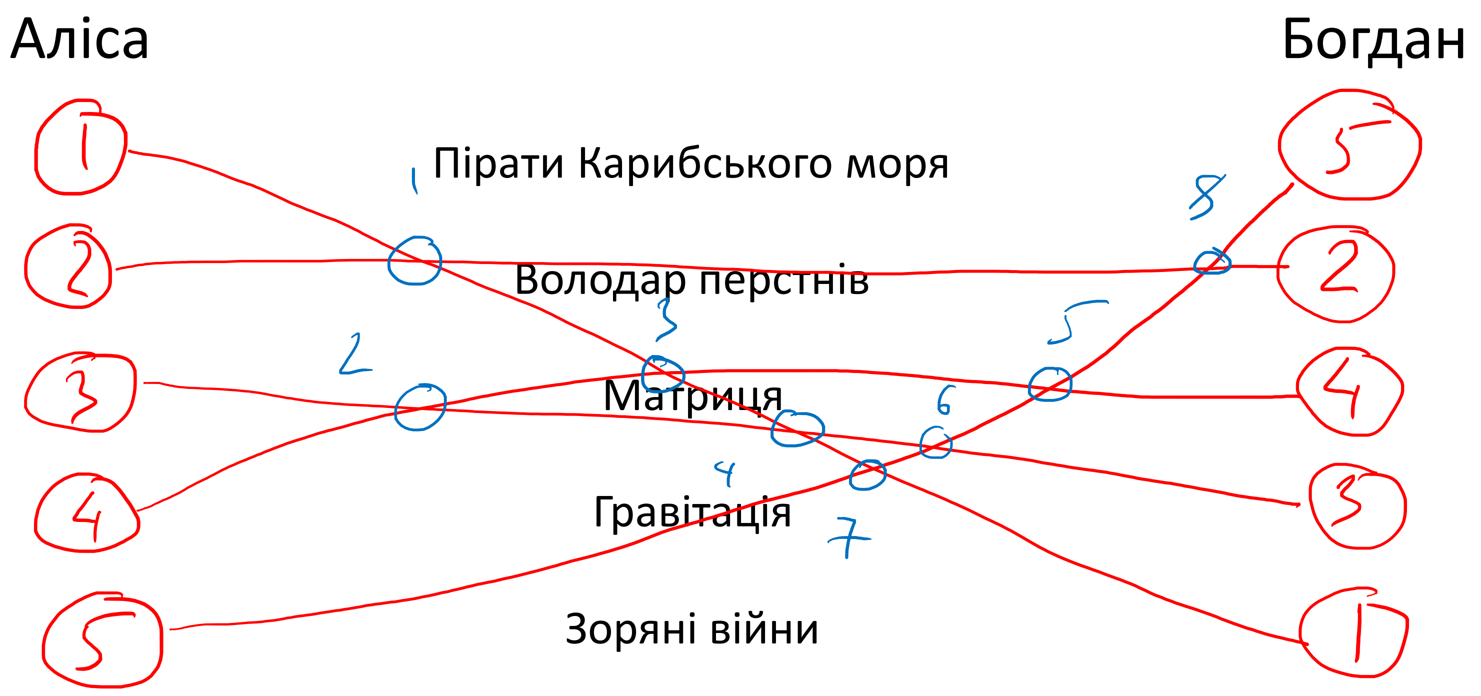
$A =$	1	2	3	6	8

$K = 0$

$A =$	8	6	3	2	1

$$K = 4 + 3 + 2 + 1 = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = 10$$

Задача порівняння вподобань



$$A = \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \hline \multicolumn{1}{|c|}{5} & \multicolumn{1}{c|}{2} & \multicolumn{1}{c|}{4} & \multicolumn{1}{c|}{3} & \multicolumn{1}{c|}{1} \\ \hline \end{array}$$

$$K = 4 + 1 + 2 + 1 = 8$$

$$n = 5, \; K_{max} = \frac{5 \cdot 4}{2} = 10$$

Метод визначення кількості інверсій

Очевидне рішення: перебір всіх пар елементів $\Rightarrow \Theta(n^2)$

<u>Метод декомпозиції</u>. Розіб'ємо масив на дві частини. (i,j):

$A = [1 \; \ldots \; \vdots \; \ldots \; n]$

1) Ліві інверсії: $A = [1 \; i \; j \; \vdots \; n]$ $\quad 1 \le i < j \le n/2$

2) Праві інверсії: $A = [1 \; \vdots \; i \; j \; n]$ $\quad n/2 < i < j$

3) Розділені інверсії: $A = [1 \; i \; \vdots \; j \; n]$ $\quad i \le n/2 < j$

$$\begin{array}{ccccccc} & 1 & 2 & 3 & 4 & 5 \\ A = & 2 & 3 & 8 & 6 & 1 \end{array}$$

$K_L = 0$

$K_R = 2 + 1 = 3$

$K_S = 2$

Алгоритм визначення кількості інверсій

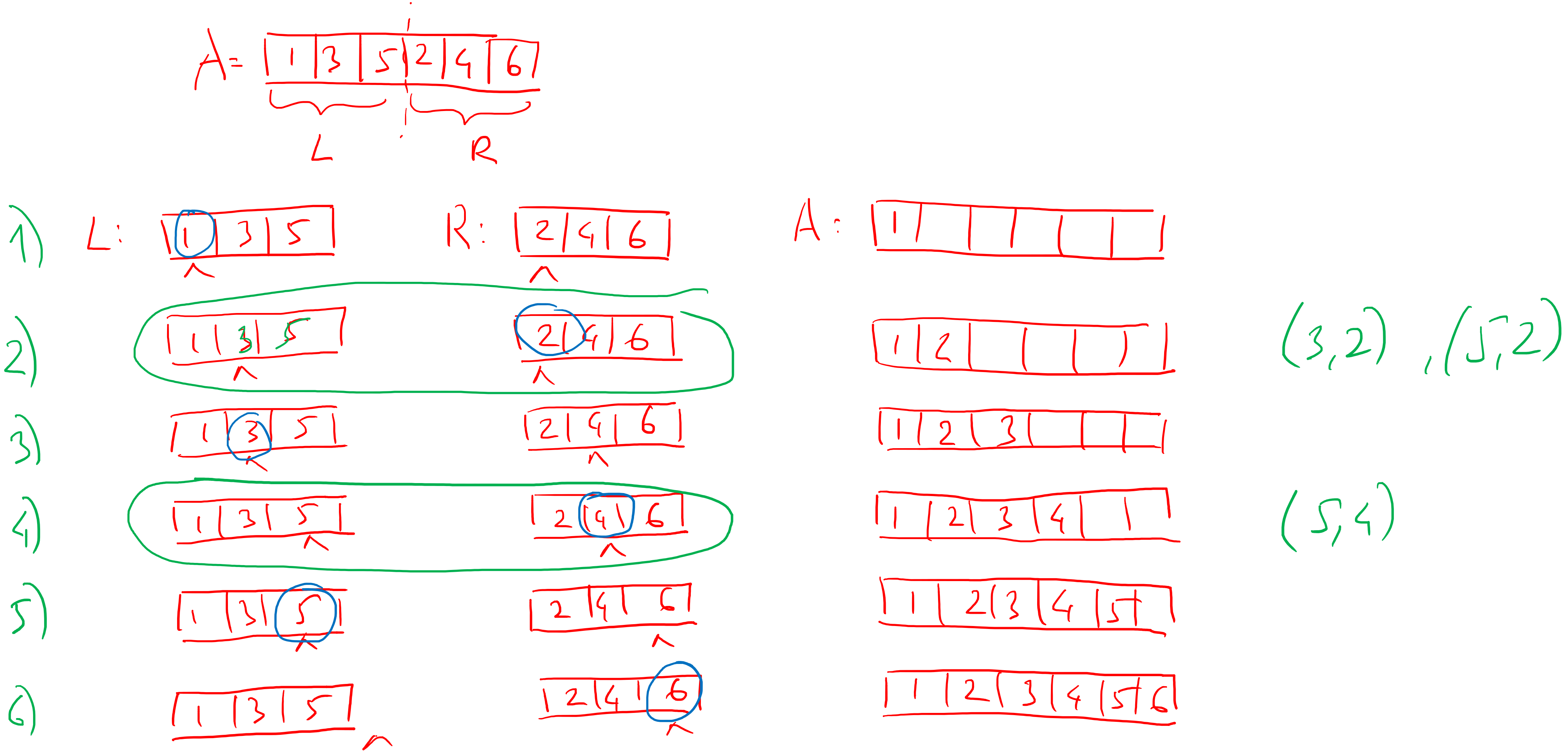
Вхід: масив чисел A

Вихід: кількість інверсій в A

CountInv(A)

1. if length(A)=1 then:
2. return 0
3. else:
4. x=CountInv(ліва частина A)
5. y=CountInv(права частина A)
6. z=CountSplitInv(A)
7. return x+y+z

Інверсії у частково відсортованому масиві



Алгоритм визначення кількості інверсій

Вхід: масив чисел A

Вихід: відсортований масив A, кількість інверсій в A

SortAndCountInv(A)

1. if length(A)=1 then:
2. return A, 0
3. else:
4. L, x = SortAndCountInv(ліва частина A)
5. R, y = SortAndCountInv(права частина A)
6. A, z = MergeAndCountSplitInv(A, L, R)
7. return A, x+y+z

Алгоритм визначення кількості інверсій

MergeAndCountSplitInv(A, L, R) ⇒ Час: $\Theta(n)$

1. $i = 1, j = 1$
2. $c = 0$ ← кількість розділених інверсій
3. for $k = 1$ to length(A) + length(R): Час SortAndCountInv : $\Theta(n \cdot \lg n)$
4. if $L[i] \leq R[j]$ then :
5. $A[k] = L[i]$
6. $i = i + 1$
7. else :
8. $A[k] = R[j]$
9. $j = j + 1$
10. $c = c + (\text{length}(L) - i + 1)$
11. return (A, c)

Задача пошуку добутку матриць

Вхід: дві матриці X та Y розмірності n×n

Вихід: $Z = X \cdot Y$, розмірність n×n $\Rightarrow$ кількість елементів: n^2

$$\Downarrow$$

$$z_{ij} = \sum_{k=1}^{n} x_{ik} \cdot y_{kj} = \underbrace{x_{i1} \cdot y_{1j} + x_{i2} \cdot y_{2j} + \ldots + x_{in} \cdot y_{nj}}_{n} \Longrightarrow \Theta(n^3)$$

Метод декомпозиції для добутку матриць

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

$$Z = X \cdot Y = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

Загальна кількість добутків: 8 розмірності $n/2 \times n/2$

$$\boxed{T(n) = 8 \cdot T(n/2) + \Theta(n^2)} \Longrightarrow T(n) = \Theta(n^3)$$

Метод Штрассена , 1969

$P_1 = A(F-H)$

$P_2 = (A+B)H$

$P_3 = (C+D)E$

$P_4 = D(G-E)$

$P_5 = (A+D)(E+H)$

$P_6 = (B-D)(G+H)$

$P_7 = (A-C)(E+F)$

$$Z = X \cdot Y = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$P_5 + P_4 - P_2 + P_6 = (A+D)(E+H) + D(G-E) - (A+B)H + (B-D)(G+H) =$$

$$= AE + \cancel{AH} + \cancel{DE} + \cancel{DH} + \cancel{DG} - \cancel{DE} - \cancel{AH} - \cancel{BH} + BG + \cancel{BH} - \cancel{DG} - \cancel{DH} = \boxed{AE + BG}$$

$$T(n) = 7 \cdot T(n/2) + \Theta(n^2) \Rightarrow O(n^{\log_2 7}) \approx O(n^{2,81})$$

1987, Coppersmith-Winograd $\Rightarrow O(n^{2,375477})$

$O(n^{2,3727})$