МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА «ПРОГРАМНА ІНЖЕНЕРІЯ ТА
ІНТЕЛЕКТУАЛЬНІ ТЕХНОЛОГІЇ УПРАВЛІННЯ»

Індивідуальне розрахункове завдання (варіант №3)
З ДИСЦИПЛІНИ «ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ»

ВИКОНАВ
Студент групи КН-221в
Шулюпов Є.Р.

ПЕРЕВІРИЛА
Проф. кафедри ПІІТУ
Стратієнко Н. К.

Харків 2022

Завдання 1

1. Визначити складність алгоритмів, записаних у вигляді псевдокоду:

а)
```
for j := 1 to n do
        for i := 1 to n - 1 do
            if a[i] > a[i + 1] then begin
                t := a[i];
                a[i] := a[i + 1];
                a[i + 1] := t;
            end;
```

б)
```
d := b * b - 4 * a * c;
s_d := sqrt(d);
x1 := (-b - s_d) / (2 * a);
x2 := (-b + s_d) / (2 * a).
```

2. Порівняти попарно швидкості росту заданих функцій, визначивши, чи можна дану пару записати, як $f(n) = O(g(n))$, при цьому $n, k > 1$:

$f_1(n) = log^k n,$

$f_2(n) = 3^k.$

3. Довести дані твердження, користуючись визначенням O-позначення:

$n^2 + 2n = O(n^2),$

$n^{45} = O(n!).$

Розв'язання

1.

Пункт (а)

№ рядка	Алгоритм	Кількість операцій (вартість)	Скільки разів буде виконаний рядок
1	`for j:=1 to n do`	c_1	$n+1$
2	`  for i:=1 to n-1 do`	c_2	n^2
3	`    if a[i] > a[i+1] then begin`	c_3	$n(n-1)$
4	`      t := a[i];`	c_4	$\leq n(n-1)$
5	`      a[i] := a[i + 1];`	c_5	$\leq n(n-1)$
6	`      a[i + 1] := t;` `    end;`	c_6	$\leq n(n-1)$

Перший рядок виконуватиметься (n + 1) разів, т.я після виконання n інтерацій ми повернемося, щоб перевірити умову. Другий рядок буде виуонаний n(зовнішній цикл) * (n – 1 + 1) = n * n = n^2 разів. Для кожного i від 1 до n – 1 підраховуємо, скільки разів буде виконано рядки 3 – 6.

$$T(n) = c_1(n+1) + c_2n^2 + (c_3 + c_4 + c_5 + c_6)n(n-1)$$

Перетворимо даний вираз, відкинувши члени менших порядків і коефіцієнт при n. Тоді отримаємо $T(n) = O(n^2)$.

Пункт (б)

d := b * b - 4 * a * c;

s_d := sqrt(d);

x1 := (-b - s_d) / (2 * a);

x2 := (-b + s_d) / (2 * a).

Даний алгоритм має постійну складність O(1).

2. $f_1(n) = log^k n,\ f_2(n) = 3^k, n, k > 1$

Якщо k – непарне число, то функція $f_1(n) = log^k n$ може бути від'ємною, а за визначенням $O(n)$ функції мають бути невід'ємні, а значить, функції $f_1(n)$ і $f_2(n)$ не можна порівнювати при непарному значенні k.

А при парних k – $log^k n \geq 0$, значить $log^k n = \mathrm{O}(3^k)$.

3.1 $n^2 + 2n = O(n^2)$

Позначимо ліву частину як $f(n) = n^2 + 2n$, а праву $g(n) = n^2$.

Запис $f(n) = \mathrm{O}(g(n))$ означає, що знайдеться така константа $c > 0$ і таке n_0, що $0 \leq f(n) \leq cg(n)$ для всіх $n \geq n_0$.

Отже, треба вказати такі константи, щоб виконувалось: $0 \leq n^2 + 2n \leq cn^2$.

Розділимо на n^2: $0 \leq 1 + \frac{2}{n} \leq$ с. Тепер легко вибрати константи, при яких даний вираз залишається вірним. Наприклад, $n_0 = 2,\ c = 2$. При $n \geq 2$ вираз $1 + \frac{2}{n}$ зменшується, значить твердження $n^2 + 2n = O(n^2)$ є вірним.

3.1 $n^{45} = O(n!)$.

n! завжди зростає швидше. Також можна обчислити $\lim_{n\to\infty} \frac{n^{45}}{n!}$. Ми отримаємо 0. Це означає, що $g(n)$ зростає швидше ніж $f(n)$. Отже, твердження є вірним.

Завдання 2

Визначити найбільшу спільну підпослідовність:

1) $addaadfbcc$ та $cdabdfadcc$;

2) $dfaedfcfba$ та $abbaedbfba$.

Розв`язання

1) Спочатку побудуємо таблицю, яка містить $(n+1)$ стовпців (з номерами від 0 до n), $(m+1)$ строк (з номерами від 0 до m), де n – число членів послідовності X, m – число членів послідовності Y.

У відповідності до рекурентних формул запишемо нулі в нульовий стовпчик та нульову строку таблиці.

	j	0	1	2	3	4	5	6	7	8	9	10
i		Y_j	c	d	a	b	d	f	a	d	c	c
0	X_i	0	0	0	0	0	0	0	0	0	0	0
1	a	0										
2	d	0										
3	d	0										
4	a	0										
5	a	0										
6	d	0										
7	f	0										
8	b	0										
9	c	0										
10	c	0										

Далі заповнюємо таблицю використовуючи рекурентне співвідношення:

$$c[i,j] = \begin{cases} 0 & \text{якщо } i = 0 \text{ чи } j = 0, \\ c[i-1, j-1] + 1, & \text{якщо } i,j > 0 \text{ та } x_i = y_j, \\ max\{c[i, j-1], c[i-1, j]\}, & \text{якщо } i,j > 0 \text{ та } x_i \neq y_j, \end{cases}$$

А також обираємо стрілку, що вказує на клітинку:

– $a[i-1, j-1]$, якщо $x_i = y_j$;

– $a[i-1, j]$, якщо $x_i \neq y_j$ і $c[i-1, j] \geq c[i, j-1]$;

– $a[i, j-1]$ якщо $x_i \neq y_j$ і $c[i-1, j] < c[i, j-1]$.

	j	0	1	2	3	4	5	6	7	8	9	10
i		Y_j	c	d	a	b	d	f	a	d	c	c
0	X_i	0	0	0	0	0	0	0	0	0	0	0
1	a	0	↑0	↑0	↖1	←1	←1	←1	↖1	←1	←1	←1
2	d	0	↑0	↖1	↑1	↑1	↖2	←2	←2	↖2	←2	←2
3	d	0	↑0	↖1	↑1	↑1	↖2	↑2	↑2	↖3	←3	←3
4	a	0	↑0	↑1	↖2	←2	↑2	↑2	↖3	↑3	↑3	↑3
5	a	0	↑0	↑1	↖2	↑2	↑2	↑2	↖3	↑3	↑3	↑3
6	d	0	↑0	↖1	↑2	↑2	↖3	←3	↑3	↖4	←4	←4
7	f	0	↑0	↑1	↑2	↑2	↑3	↖4	←4	↑4	↑4	↑4
8	b	0	↑0	↑1	↑2	↖3	↑3	↑4	↑4	↑4	↑4	↑4
9	c	0	↖1	↑1	↑2	↑3	↑3	↑4	↑4	↑4	↖5	↖5
10	c	0	↖1	↑1	↑2	↑3	↑3	↑4	↑4	↑4	↖5	↖6

Коли таблиця повністю заповнені, піднімаємося за стрілками, починаючи з нижньої правої клітинки. Цифра 6 у нижній правій клітинці вказує на кількість елементів підпослідовності. Записуємо до нашої підпослідовності тільки ті клітинки, в яких стрілка вказує на верхній лівий кут: (adadcc).

2) Спочатку побудуємо таблицю, яка містить $(n+1)$ стовпців (з номерами від 0 до n), $(m+1)$ строк (з номерами від 0 до m), де n – число членів послідовності X, m – число членів послідовності Y.

У відповідності до рекурентних формул запишемо нулі в нульовий стовпчик та нульову строку таблиці.

	j	0	1	2	3	4	5	6	7	8	9	10
i		Y_j	a	b	b	a	e	d	b	f	b	a
0	X_i	0	0	0	0	0	0	0	0	0	0	0
1	d	0										
2	f	0										
3	a	0										
4	e	0										
5	d	0										
6	f	0										
7	c	0										
8	f	0										
9	b	0										
10	a	0										

Далі заповнюємо таблицю використовуючи рекурентне співвідношення:

$$c[i,j] = \begin{cases} 0 & \text{якщо } i = 0 \text{ чи } j = 0, \\ c[i-1, j-1] + 1, & \text{якщо } i, j > 0 \text{ та } x_i = y_j, \\ max\{c[i, j-1], c[i-1, j]\}, & \text{якщо } i, j > 0 \text{ та } x_i \neq y_j, \end{cases}$$

А також обираємо стрілку, що вказує на клітинку:

– $a[i-1, j-1]$, якщо $x_i = y_j$;

– $a[i-1, j]$, якщо $x_i \neq y_j$ і $c[i-1, j] \geq c[i, j-1]$;

– $a[i, j-1]$ якщо $x_i \neq y_j$ і $c[i-1, j] < c[i, j-1]$.

	j	0	1	2	3	4	5	6	7	8	9	10
i		Y_j	a	b	b	a	e	d	b	f	b	a
0	X_i	0	0	0	0	0	0	0	0	0	0	0
1	d	0	↑0	↑0	↑0	↑0	↑0	↖1	←1	←1	←1	←1
2	f	0	↑0	↑0	↑0	↑0	↑0	↑1	↑1	↖2	←2	←2
3	a	0	↖1	←1	←1	↖1	←1	↑1	↑1	↑2	↑2	↖3
4	e	0	↑1	↑1	↑1	↑1	↖2	←2	←2	↑2	↑2	↑3
5	d	0	↑1	↑1	↑1	↑1	↑2	↖3	←3	←3	←3	↑3
6	f	0	↑1	↑1	↑1	↑1	↑2	↑3	↑3	↖4	←4	←4
7	c	0	↑1	↑1	↑1	↑1	↑2	↑3	↑3	↑4	↑4	↑4
8	f	0	↑1	↑1	↑1	↑1	↑2	↑3	↑3	↖4	↑4	↑4
9	b	0	↑1	↖2	↖2	←2	↑2	↑3	↖4	↑4	↖5	←5
10	a	0	↖1	↑2	↑2	↖3	←3	↑3	↑4	↑4	↑5	↖6

Коли таблиця повністю заповнені, піднімаємося за стрілками, починаючи з нижньої правої клітинки. Цифра 6 у нижній правій клітинці вказує на кількість елементів підпослідовності. Записуємо до нашої підпослідовності тільки ті клітинки, в яких стрілка вказує на верхній лівий кут: (aedfba).

Завдання 3

Розв’язати задачу про заявки жадібним алгоритмом.

1)

$[12, 15);\ [1, 3);\ [14, 16);\ [4, 5);\ [10, 13);\ [7, 13);\ [8, 10);\ [5, 6);\ [7, 10);\ [5, 8);$

2)

$[1, 3);\ [4, 6);\ [7, 12);\ [13, 15);\ [13, 16);\ [8, 9);\ [11, 15);\ [7, 11);\ [11, 12);\ [3, 5).$

Розв\`язання

1) Спочатку упорядкуємо заявки за зростанням часу їх закінчення:

$$a_1 = [1, 3);\ a_2 = [4, 5);\ a_3 = [5, 6);\ a_4 = [5, 8);\ a_5 = [7, 10);\ a_6 = [8, 10);$$

$$a_7 = [7, 13);\ a_8 = [10, 13);\ a_9 = [12, 15);\ a_{10} = [14, 16);$$

Перший етап: обираємо першу заявку.

Другий етап: шукаємо заявку, що починається не раніше, ніж закінчується перша заявка. Розглядаємо заявки в порядку зростання їх номерів. Заявка a_2 задовольняє вимогам, тому вибираємо її.

Третій етап: вибираємо заявку a_3, так як вона задовольняє вимогам.

Четвертий етап: час початку заявки a_4 менший, ніж час закінчення a_3, тому виключаємо заявку a_4 з подальшого розгляду. Заявка a_5 задовольняє вимогам, тому вибираємо її.

П’ятий етап: виключаємо з подальшого розгляду заявки a_6 та a_7, так як вони починаються раніше, ніж закінчується заявка a_5. Заявка a_8 задовольняє вимогам, тому вибираємо її.

Шостий етап: заявка a_9 починається раніше, ніж закінчується заявка a_8, тому виключаємо її з подальшого розгляду. Заявка a_{10} задовольняє вимогам, додаємо її до загального розв’язку.

Таким чином, оптимальне рішення містить заявки a_1, a_2, a_3, a_5, a_8 та a_{10}.

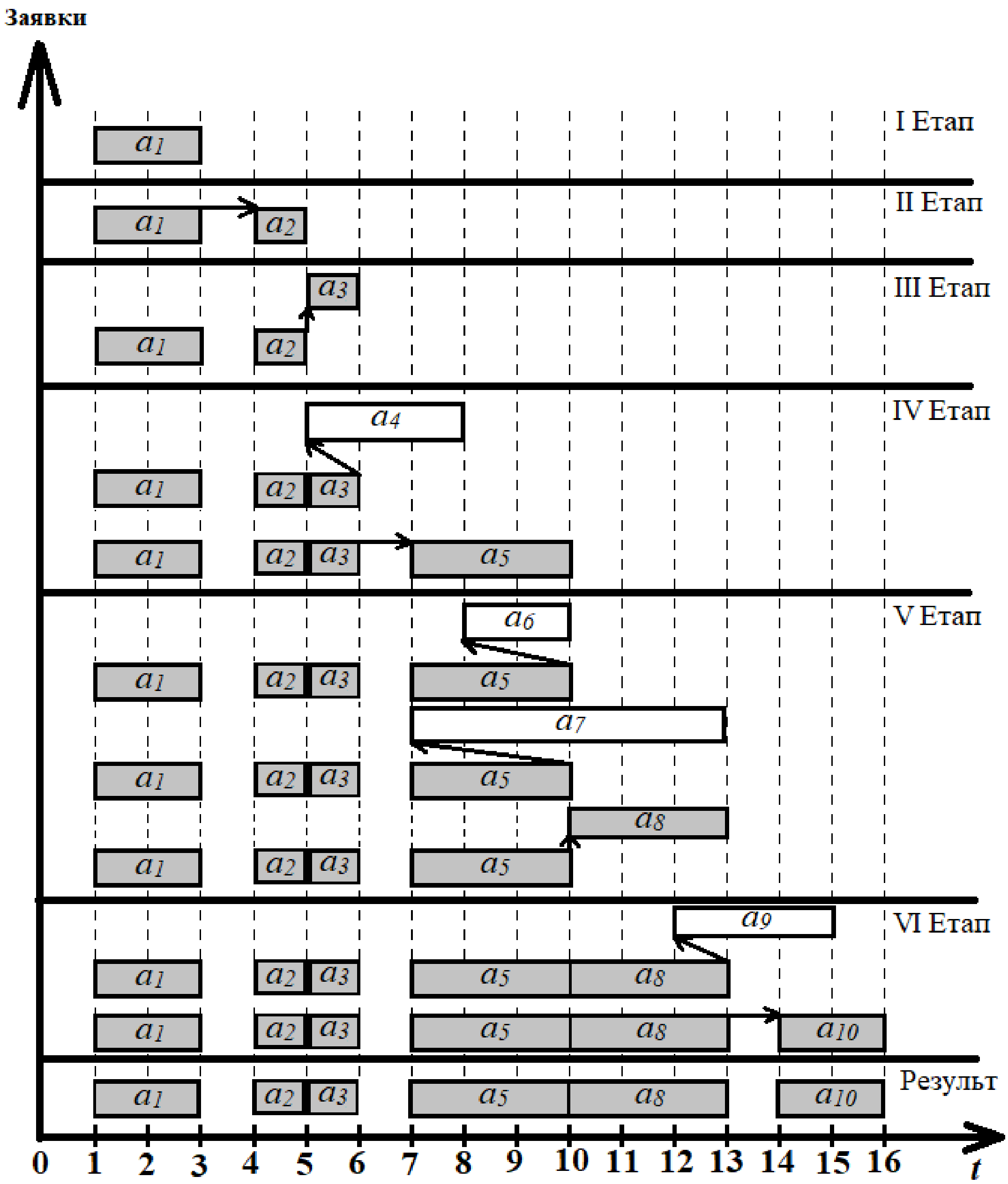


Рисунок «Ілюстрація розв’язку завдання 3.1»

2) Спочатку упорядкуємо заявки за зростанням часу їх закінчення:

$$a_1 = [1,3);\ a_2 = [3,5);\ a_3 = [4,6);\ a_4 = [8,9);\ a_5 = [7,11);\ a_6 = [7,12);$$

$$a_7 = [11,12);\ a_8 = [11,15);\ a_9 = [13,15);\ a_{10} = [13,16);$$

<u>Перший етап</u>: обираємо першу заявку.

<u>Другий етап</u>: шукаємо заявку, що починається не раніше, ніж закінчується перша заявка. Розглядаємо заявки в порядку зростання їх номерів. Заявка a_2 задовольняє вимогам, тому вибираємо її.

<u>Третій етап</u>: час початку заявки a_3 менший, ніж час закінчення a_2, тому виключаємо заявку a_3 з подальшого розгляду. Заявка a_4 задовольняє вимогам, тому вибираємо її.

<u>Четвертий етап</u>: виключаємо з подальшого розгляду заявки a_5 та a_6, так як вони починаються раніше, ніж закінчується заявка a_4. Заявка a_7 задовольняє вимогам, тому вибираємо її.

<u>П'ятий етап</u>: заявка a_8 починається раніше, ніж закінчується заявка a_7, тому виключаємо її з подальшого розгляду. Заявка a_9 задовольняє вимогам, додаємо її до загального розв'язку.

<u>Шостий етап</u>: заявка a_{10} починається раніше, ніж закінчується заявка a_9, тому виключаємо її з подальшого розгляду. Всі точки розглянуто.

Таким чином, оптимальне рішення містить заявки a_1, a_2, a_4, a_7, та a_9.

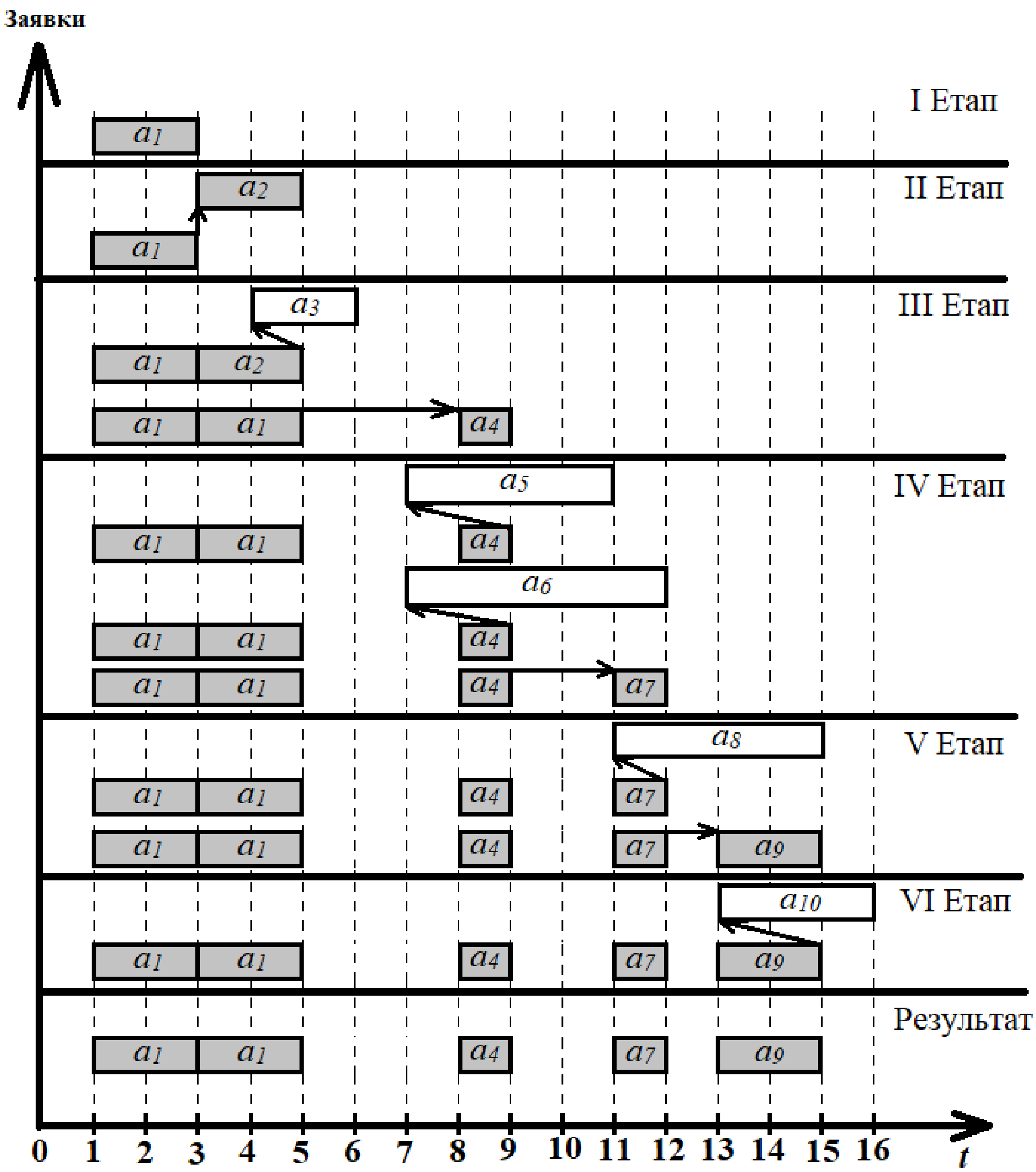


Рисунок «Ілюстрація розв'язку завдання 3.2»

Завдання 4

Написати програму для машини Тьюринга, яка додає два числа. Перевірити її роботу на числах a, b. Для визначення чисел необхідно додати 10 до номера варіанту та взяти першу цифру суми в якості a, другу – як b.

Розв`язання

Запишемо програму до таблиці, вказуючи напрямок зміщення стрічки:

Q / A	q_0	q_1	q_2	q_3
1	**q_1_L**	**$q_1$1L**	**$q_2$1R**	**STOP**
+	**q_3_S**	**q_1+L**	**q_2+R**	
_	**q_0_L**	**$q_2$1R**	**q_0_L**	

Зовнішній алфавіт: A = {1, +, _}, де “_” – пуста комірка.

Внутрішній алфавіт: Q = {q0, q1, q2, q3}, q3 – завершальний стан.

q_0

0	1	2	3	4	5	6
_	**1**	+	**1**	**1**	**1**	_

q_0_ -> q_0_L

q_0

0	1	2	3	4	5	6
_	**1**	+	**1**	**1**	**1**	_

$q_0$1 -> q_1_L

q_1

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	_

q_1+ -> q_1+L

q_1

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	_

$q_1$1 -> $q_1$1L

q_1

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	_

$q_1$1 -> $q_1$1L

q_1

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	_

$q_1$1 -> $q_1$1L

q_1

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	_

q_1_ -> $q_2$1R

q_2

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	**1**

$q_2$1 -> $q_2$1R

q_2

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	**1**

$q_2$1 -> $q_2$1R

q_2

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	**1**

$q_2$1 -> $q_2$1R

q_2

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	**1**

q_2+ -> q_2+R

q_2

0	1	2	3	4	5	6
_	_	+	**1**	**1**	**1**	**1**

q_2_ -> q_0_L

q_0

0	1	2	3	4	5	6
_	**_**	**+**	**1**	**1**	**1**	**1**

q_0+ -> q_3_S

q_3

0	1	2	3	4	5	6
_	**_**	**_**	**1**	**1**	**1**	**1**

q_3_ -> STOP

Завдання 5

Виконати сортування послідовностей:

1) 5 9 1 7 1 7 4 8 2 4;

2) 2 9 1 9 7 2 0 5 7 8;

Методи сортування:

1) Бульбашкою;

2) Кучею;

3) Злиттям;

4) Швидким сортуванням.

Розв’язання

1. Бульбашкою

1) Перший прохід:

5917174824 → 5**91**7174824 → 51**97**174824 → 517**91**74824 → 5171**97**4824 → 51717**94**824 → 517174**98**24 → 5171748**92**4 → 51717482**94** → 5171748249

Другий прохід:

5171748249 → 1**57**1748249 → 15**71**748249 → 151**77**48249 → 1517**74**8249 → 15174**78**249 → 151747**82**49 → 1517472**84**9 → 1517472489

Третій прохід:

1517472489 → 1**51**7472489 → 11**57**472489 → 115**74**72489→ 1154**77**2489 → 11547**72**489 → 115472**74**89 → 1154724789

Четвертий прохід:

1154724789 → 1**15**4724789 → 11**54**724789 → 114**57**24789 → 1145**72**4789 → 11452**74**789 → 1145247789

П’ятий прохід:

1145247789 → 1**14**5247789 → 11**45**247789 → 114**52**47789 → 1142**54**7789 → 1142457789

Шостий прохід:

1142457789 → 1**14**2457789 → 11**42**457789 → 112**44**57789 → 1124457789

Сьомий прохід:

1124457789 → 1**12**4457789 → 11**24**457789 → 1124457789

Восьмий прохід:

1124457789 → 1**12**4457789 → 1124457789

Дев’ятий прохід:

1124457789 → 1124457789

2) Перший прохід:

2919720578 → 2**91**9720578 → 21**99**720578 → 219**97**20578→ 2197**92**0578 → 21972**90**578 → 219720**95**78 → 2197205**97**8 → 21972057**98** → 2197205789

Другий прохід:

2197205789 → 1**29**7205789 → 12**97**205789 → 127**92**05789 → 1272**90**5789 → 12720**95**789 → 127205**97**89 → 1272057**98**9 →

1272057899

Третій прохід:

1272057899 → 1**27**2057899 → 12**72**057899 → 122**70**57899 → 1220**75**7899 → 12205**77**899 → 122057**78**99 → 1220577899

Четвертий прохід:

1220577899 → 1**22**0577899 → 12**20**577899 → 120**25**77899 → 1202**57**7899 → 12025**77**899 → 1202577899

П’ятий прохід:

1202577899 → 1**20**2577899 → 10**22**577899 → 102**25**77899 → 1022**57**7899 → 1022577899

Шостий прохід:

1022577899 → 0**12**2577899 → 01**22**577899 → 012**25**77899 → 0122577899

Сьомий прохід:

0122577899 → 0**12**2577899 → 01**22**577899 → 0122577899

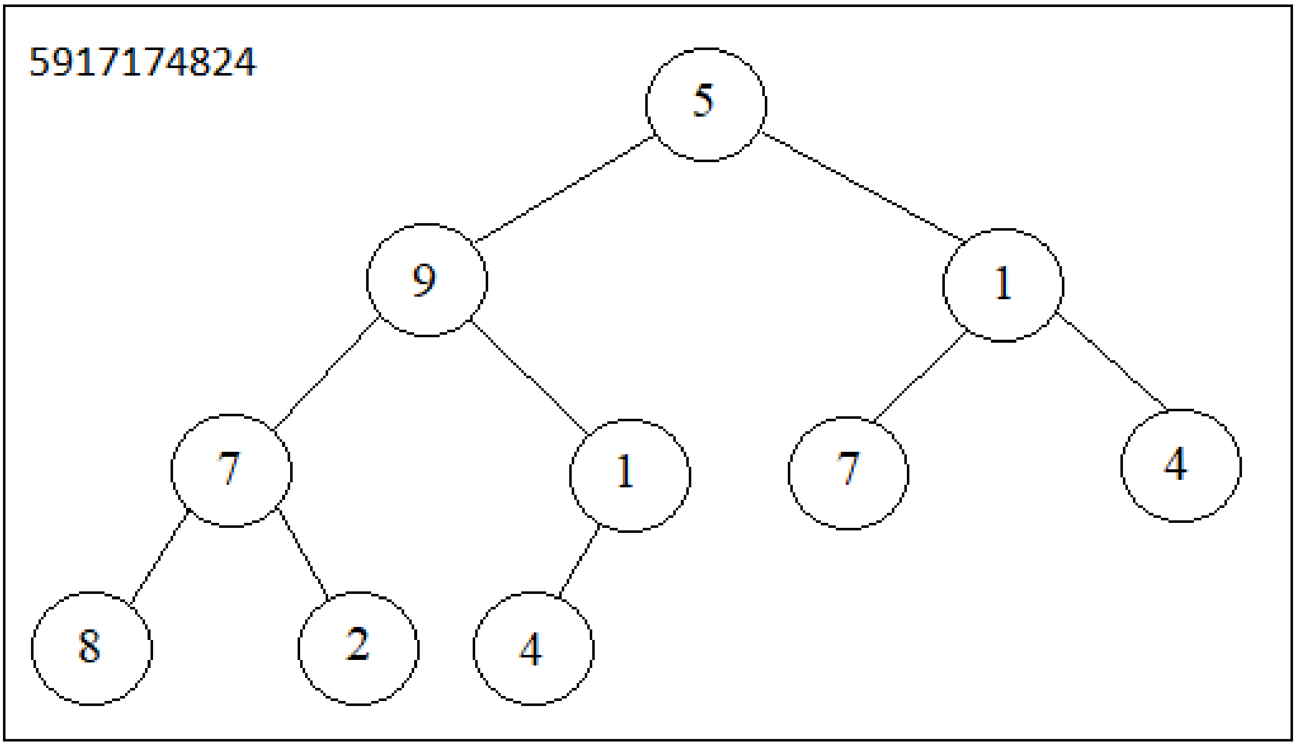
Восьмий прохід:

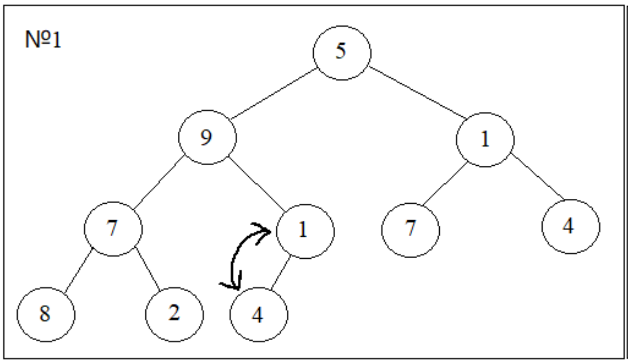
0122577899 → 0**12**2577899 → 0122577899

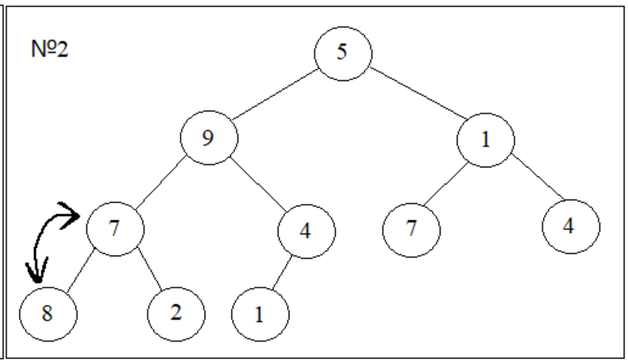
Дев’ятий прохід:

0122577899 → 0122577899

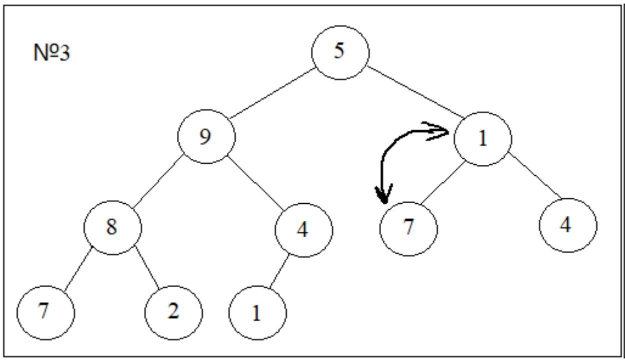
2. Кучею



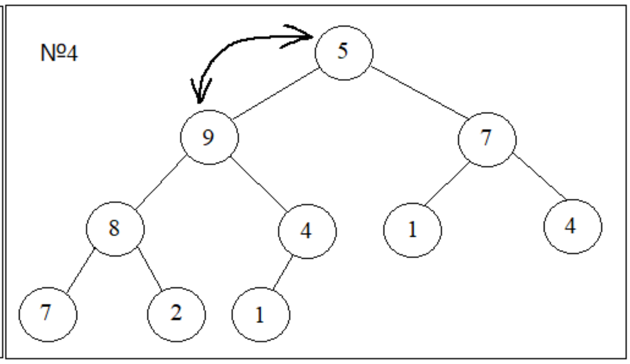




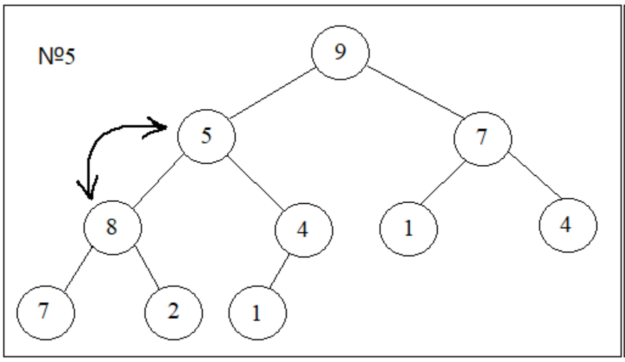
№3



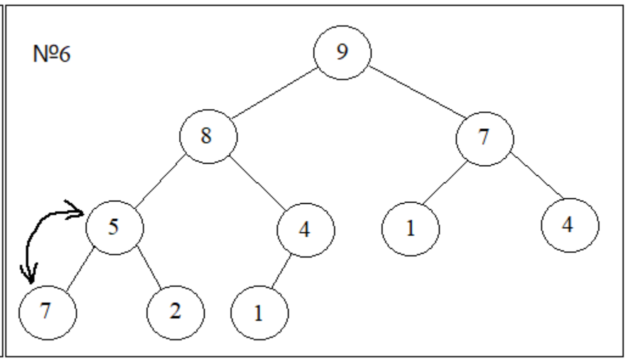
№4



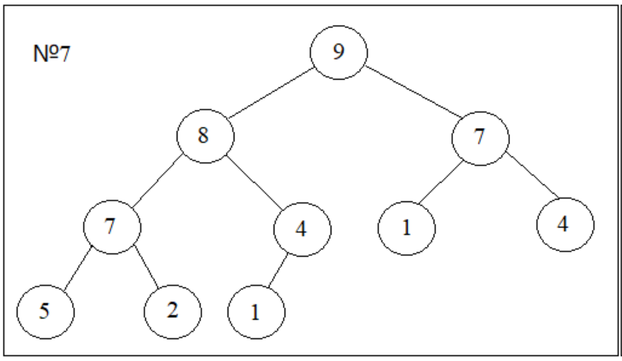
№5



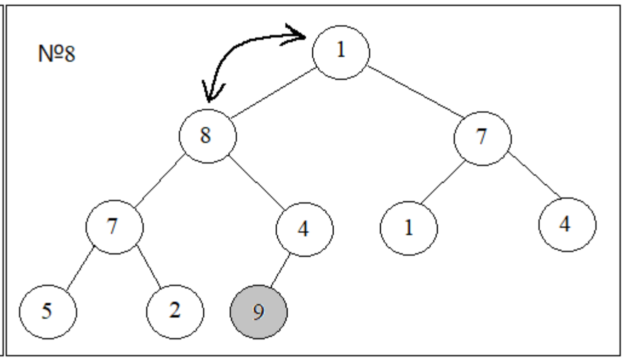
№6



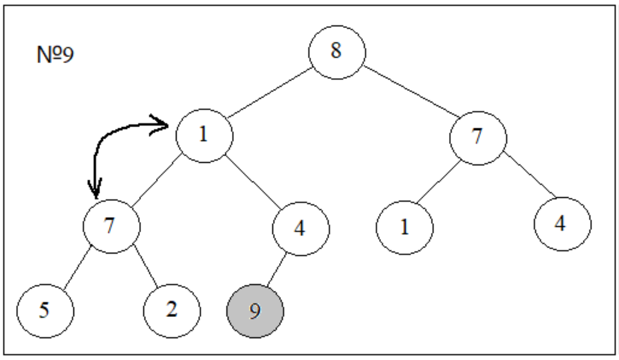
№7



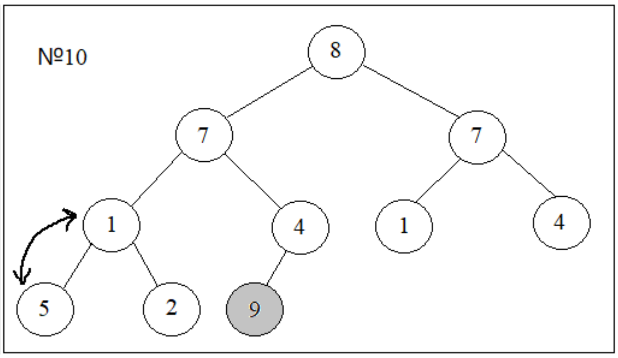
№8



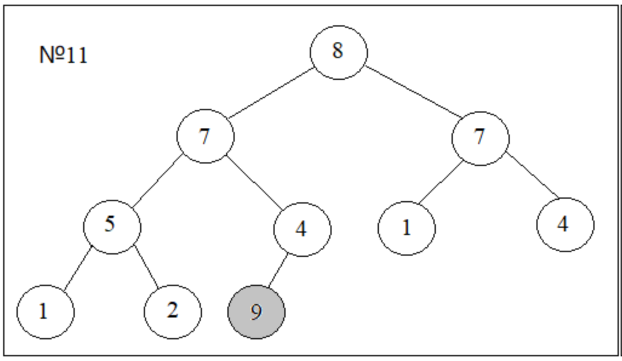
№9



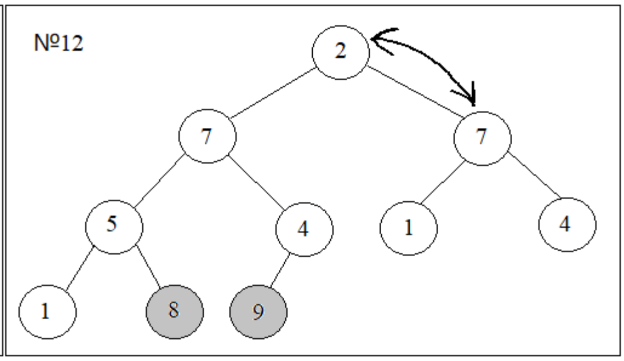
№10



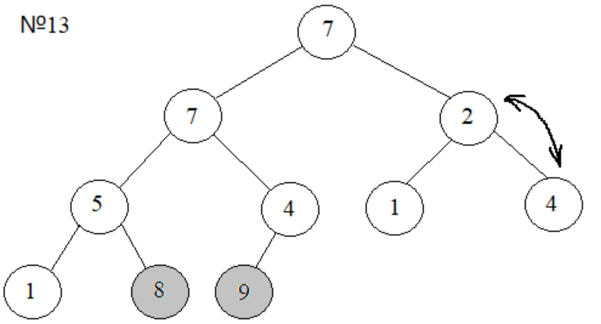
№11



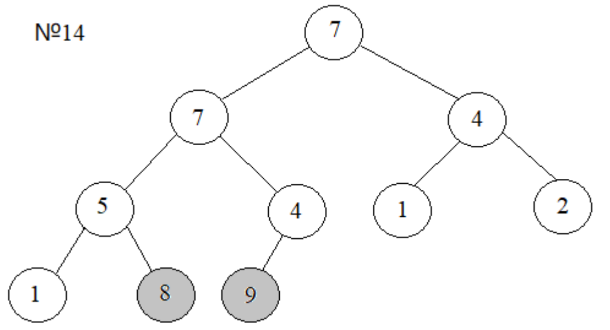
№12



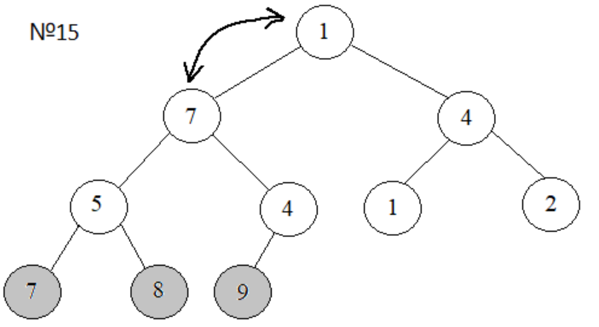
№13



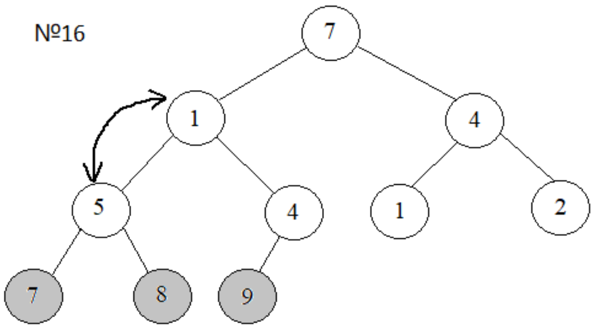
№14



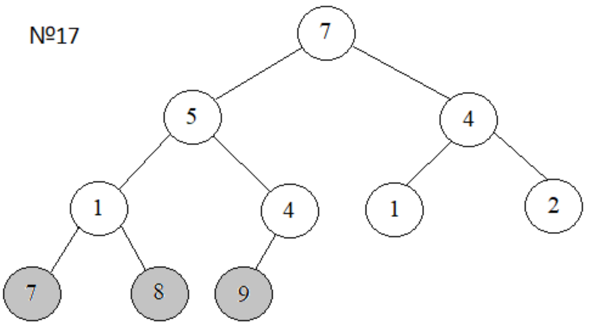
№15



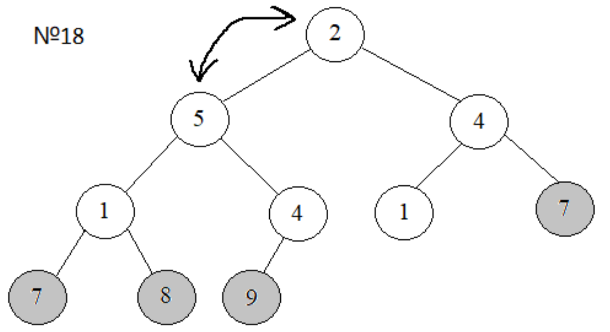
№16



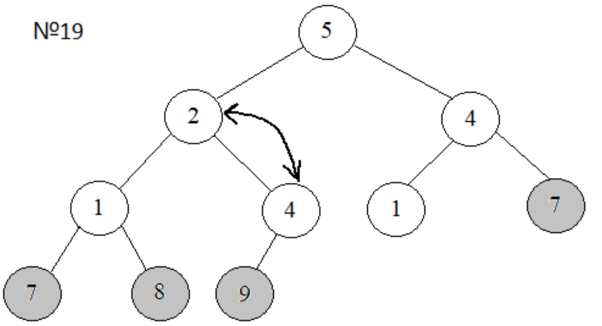
№17



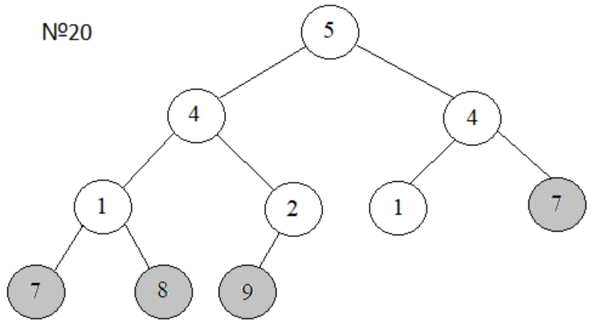
№18



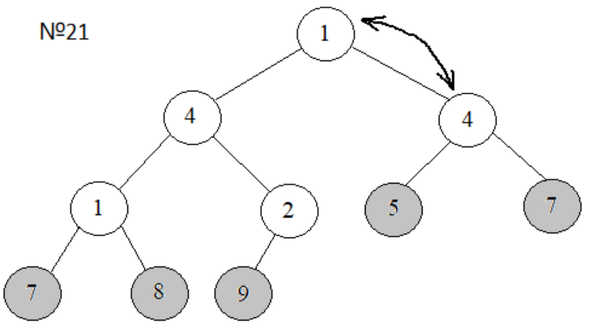
№19



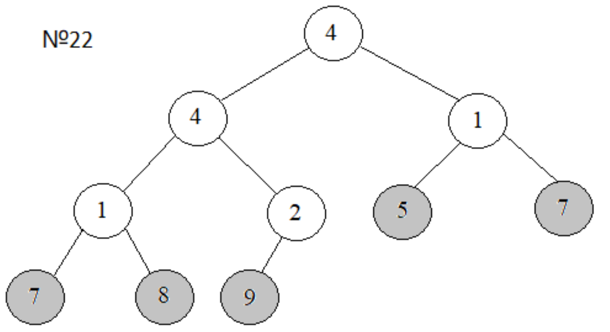
№20



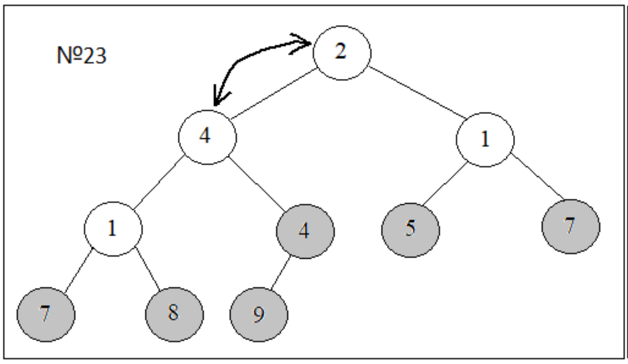
№21



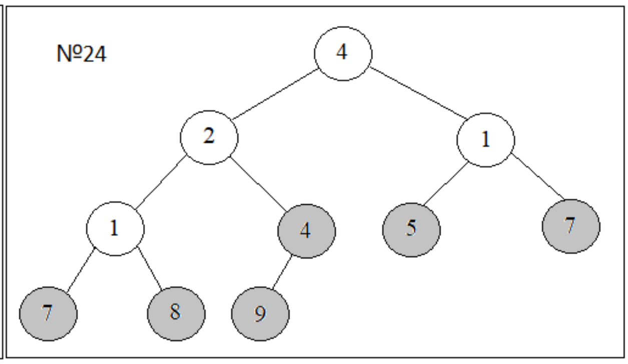
№22



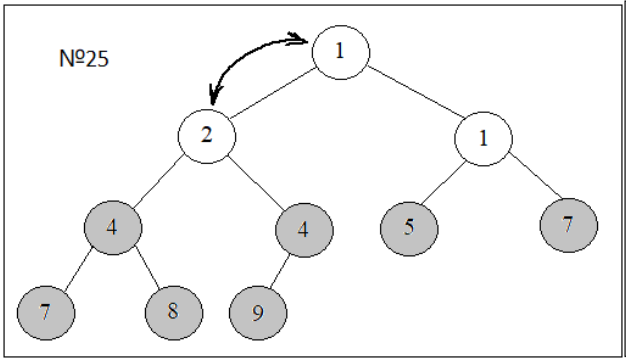
№23



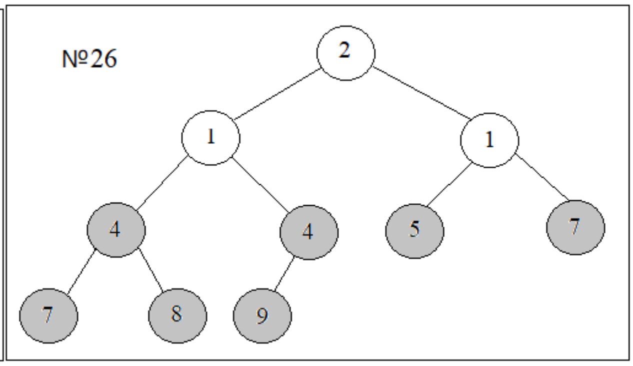
№24



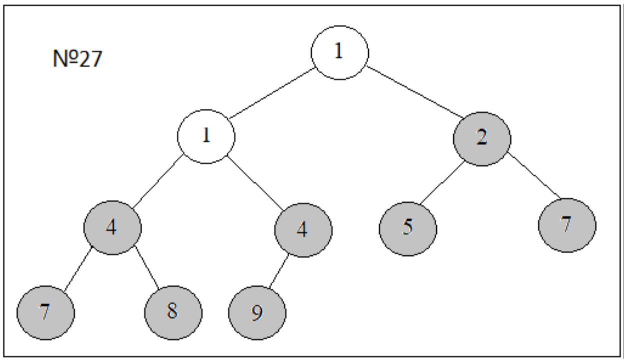
№25



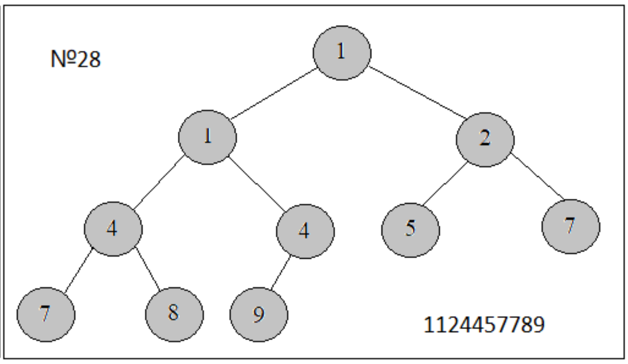
№26

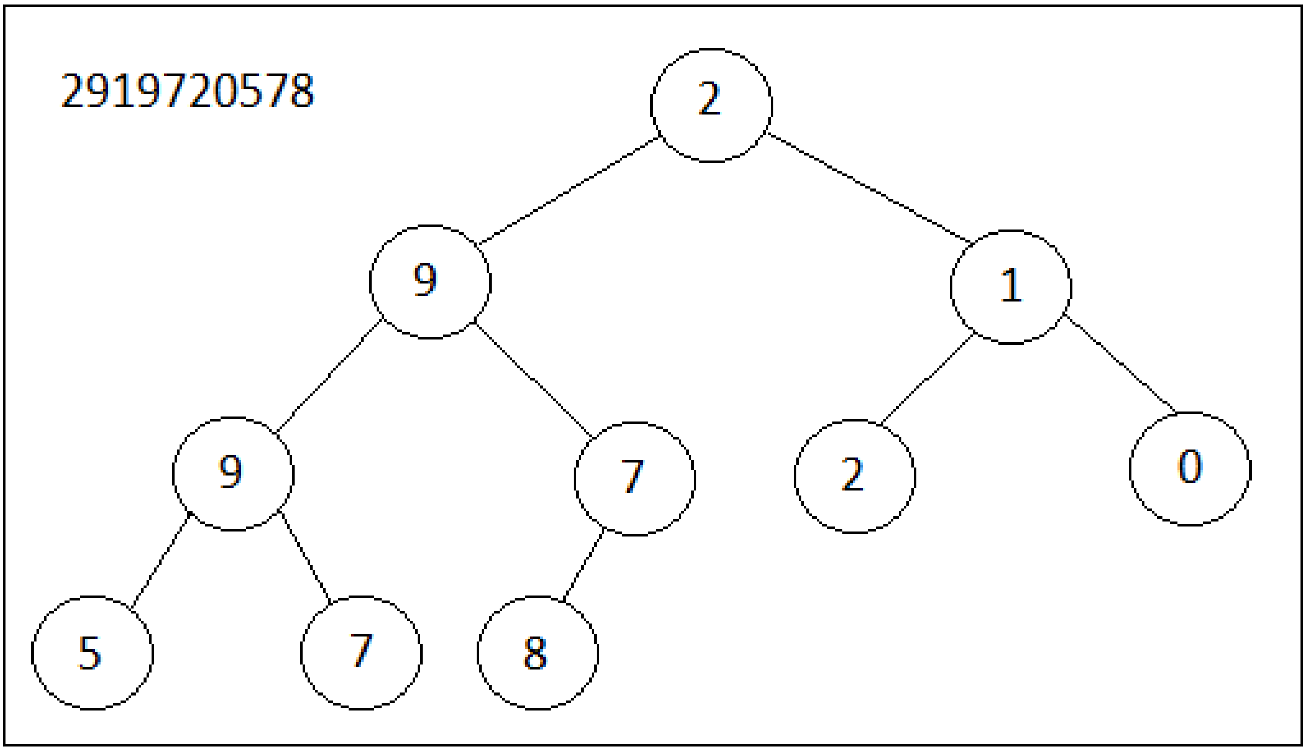


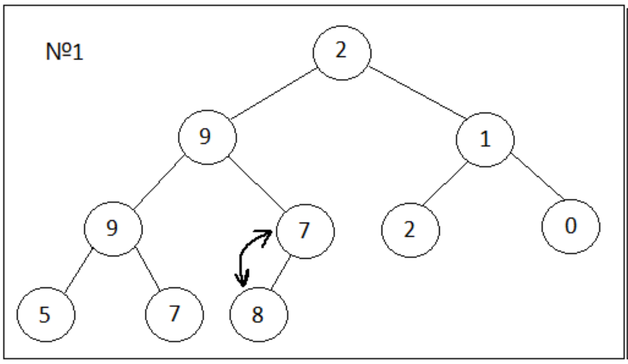
№27

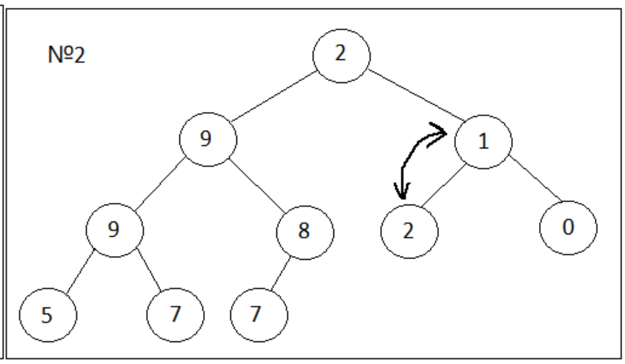


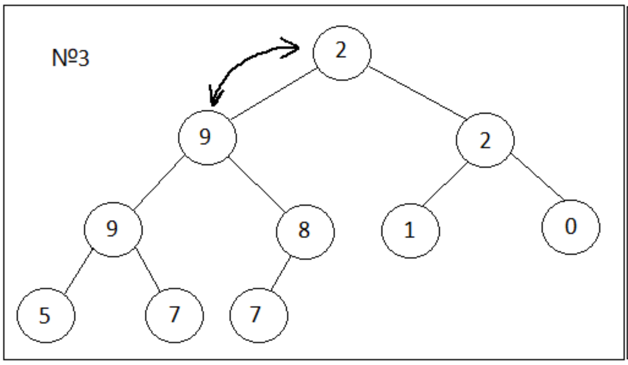
№28

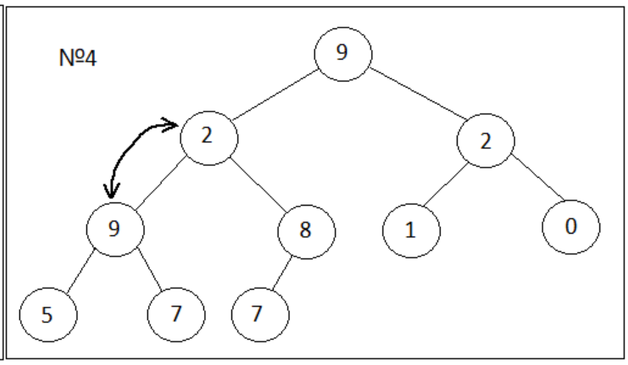


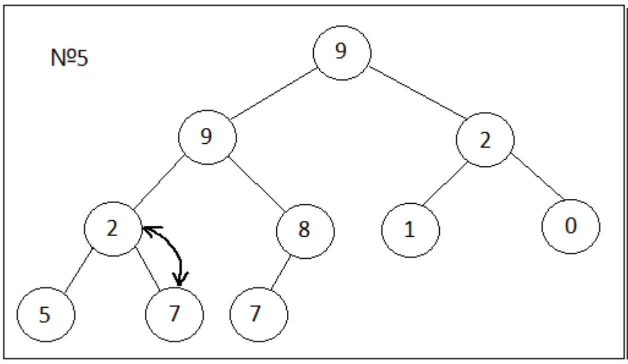

2919720578

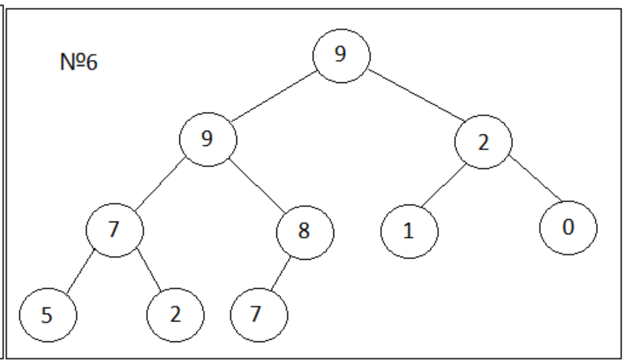

№1

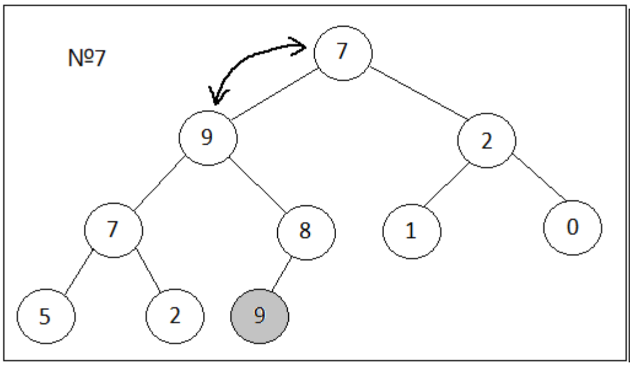

№2

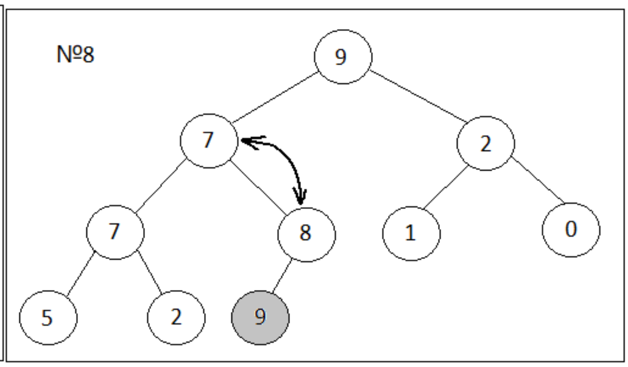

№3

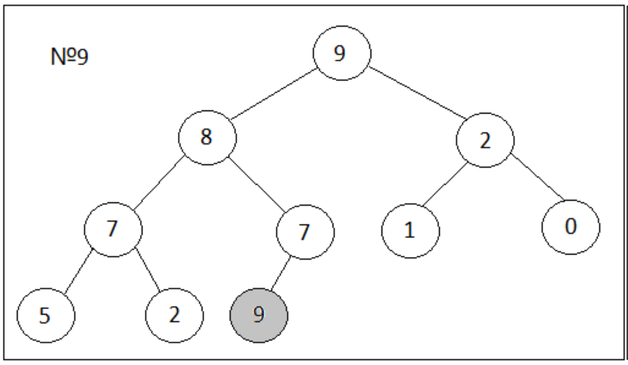

№4

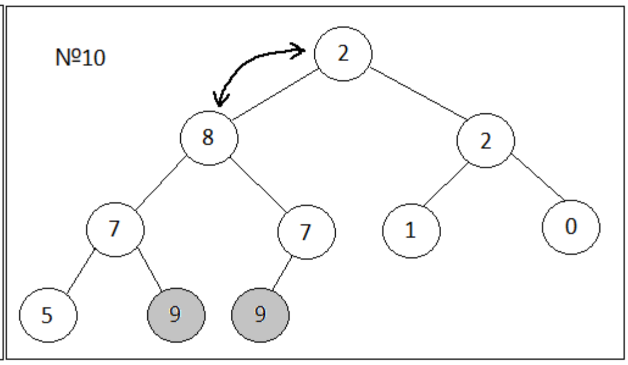

№5

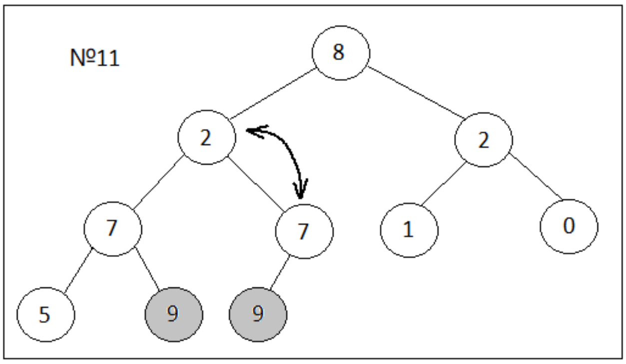

№6

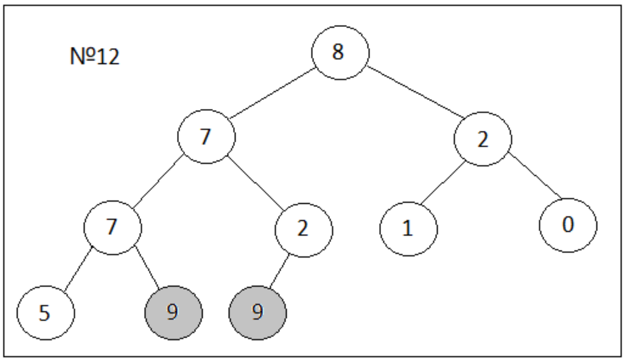

№7

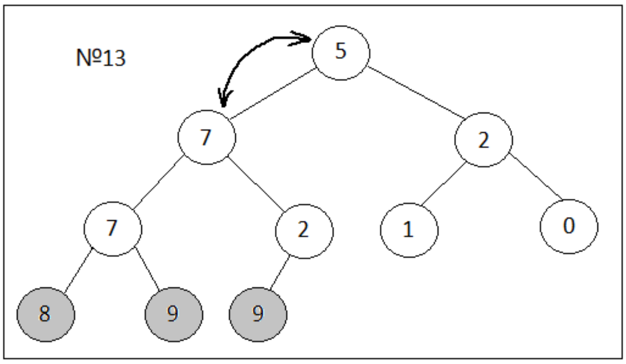

№8

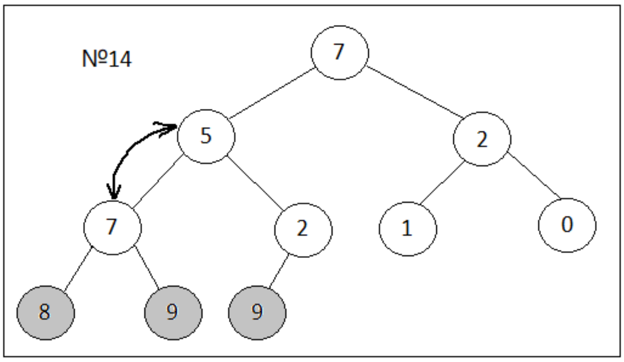

№9

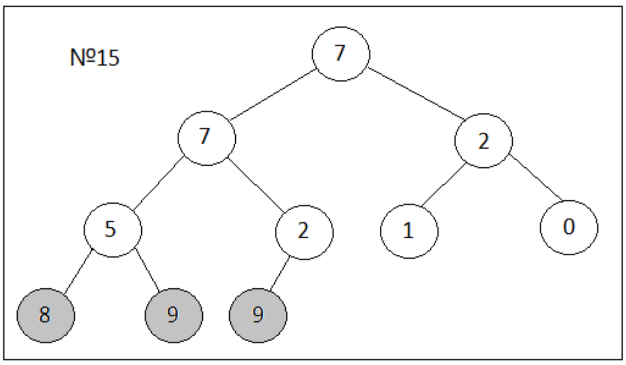

№10

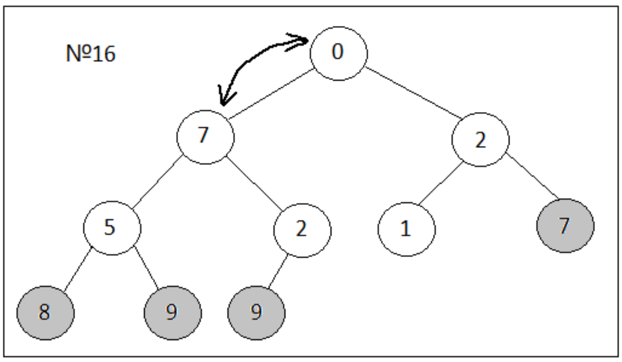

№11

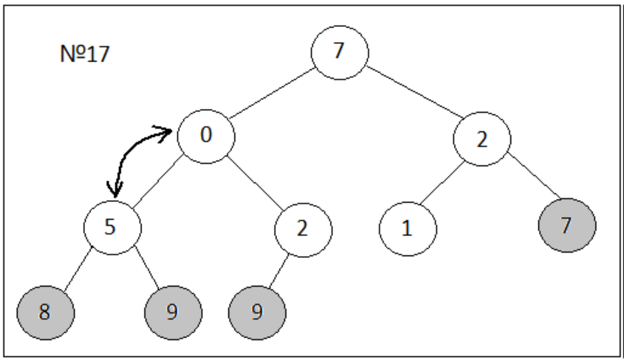

№12

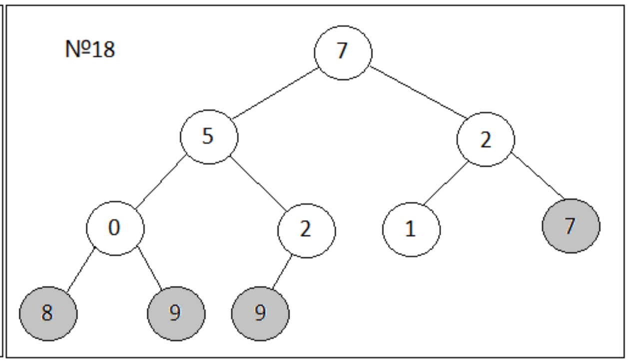

№13

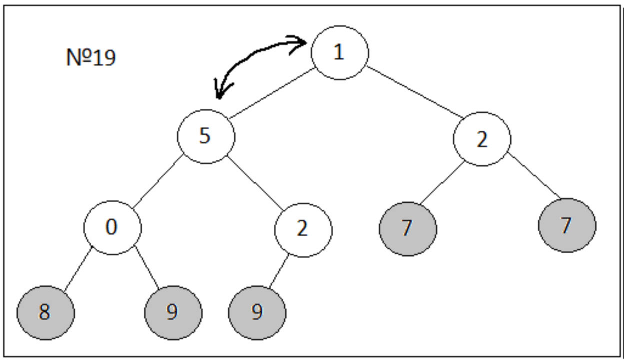

№14

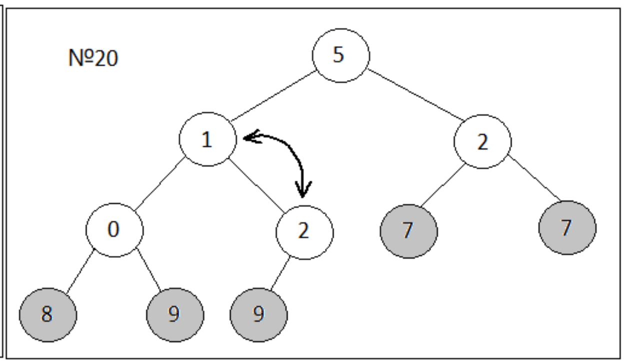

№15

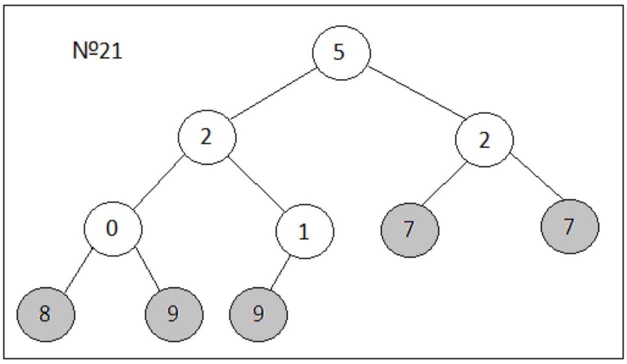

№16

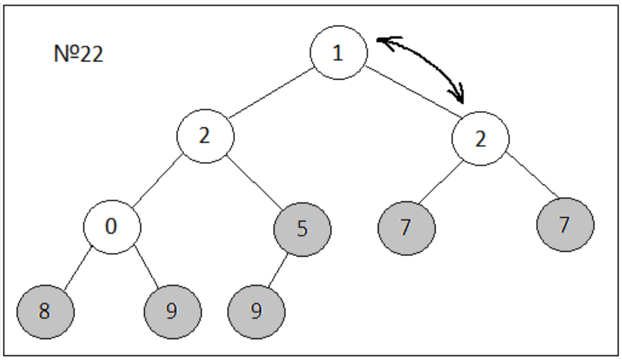

№17
7
0 2
5 2 1 7
8 9 9

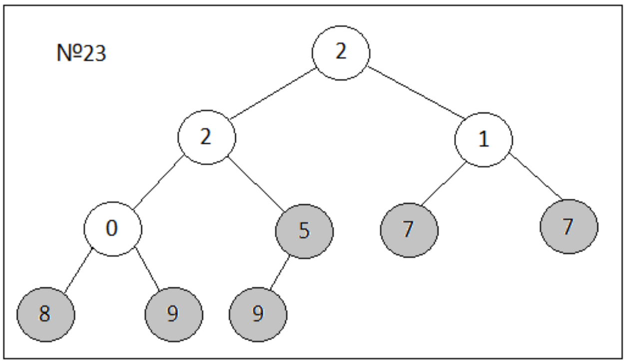

№18
7
5 2
0 2 1 7
8 9 9

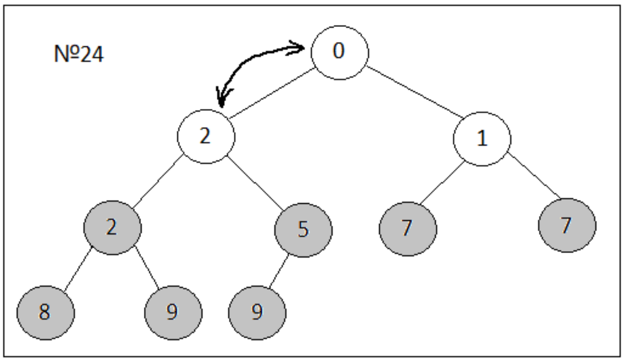

№19
1
5 2
0 2 7 7
8 9 9

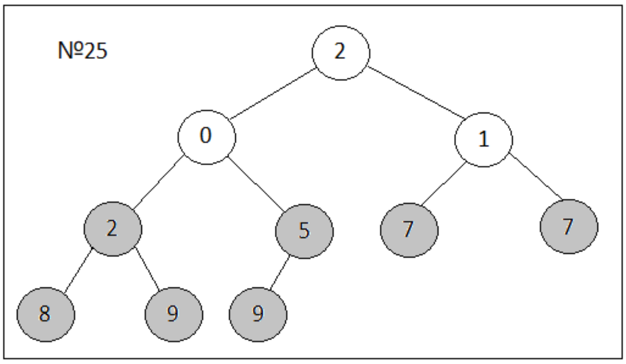

№20
5
1 2
0 2 7 7
8 9 9

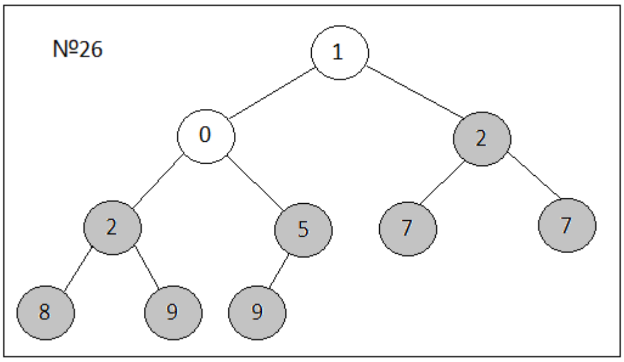

№21
5
2 2
0 1 7 7
8 9 9

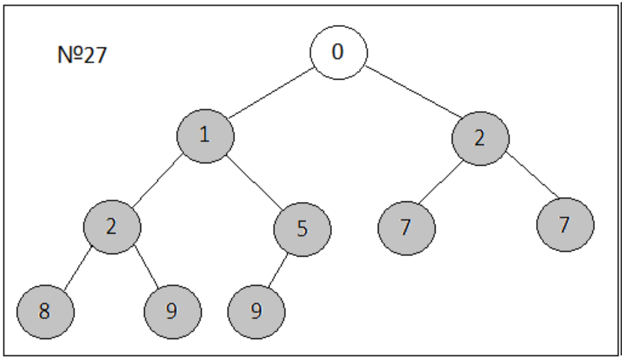

№22
1
2 2
0 5 7 7
8 9 9

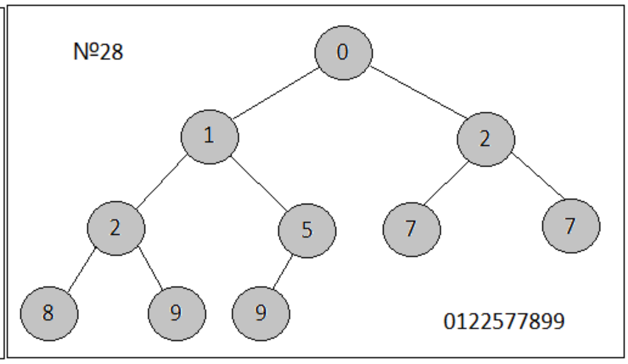

№23
2
2 1
0 5 7 7
8 9 9


№24
0
2 1
2 5 7 7
8 9 9

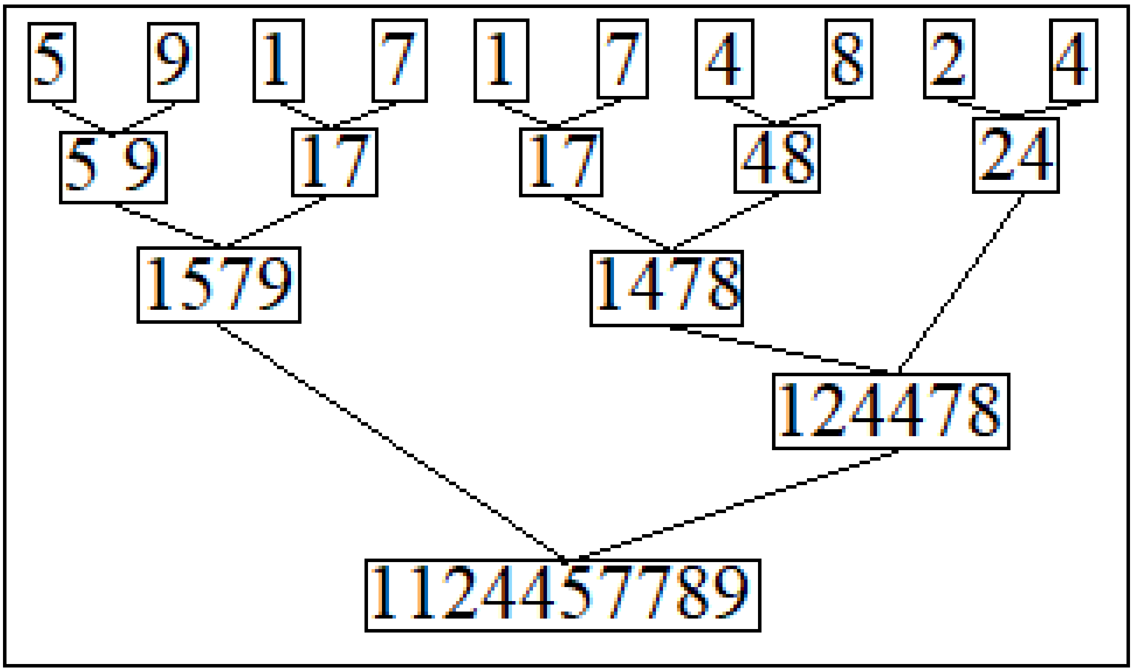

№25
2
0 1
2 5 7 7
8 9 9

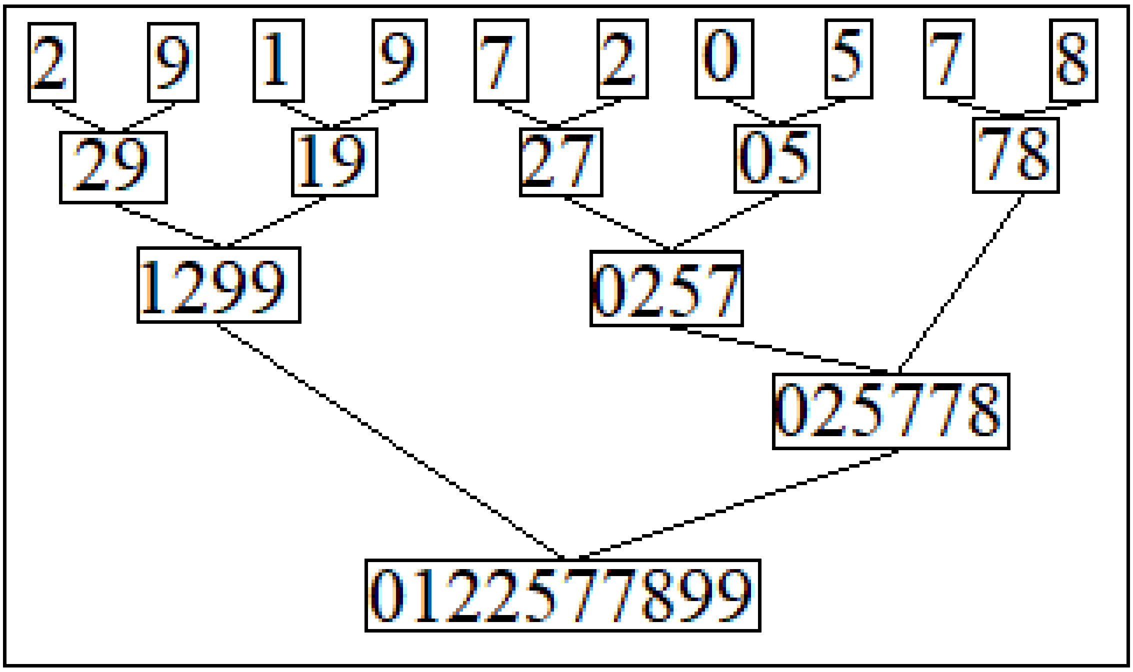

№26
1
0 2
2 5 7 7
8 9 9


№27
0
1
2
2
5
7
7
8
9
9


№28
0
1
2
2
5
7
7
8
9
9
0122577899

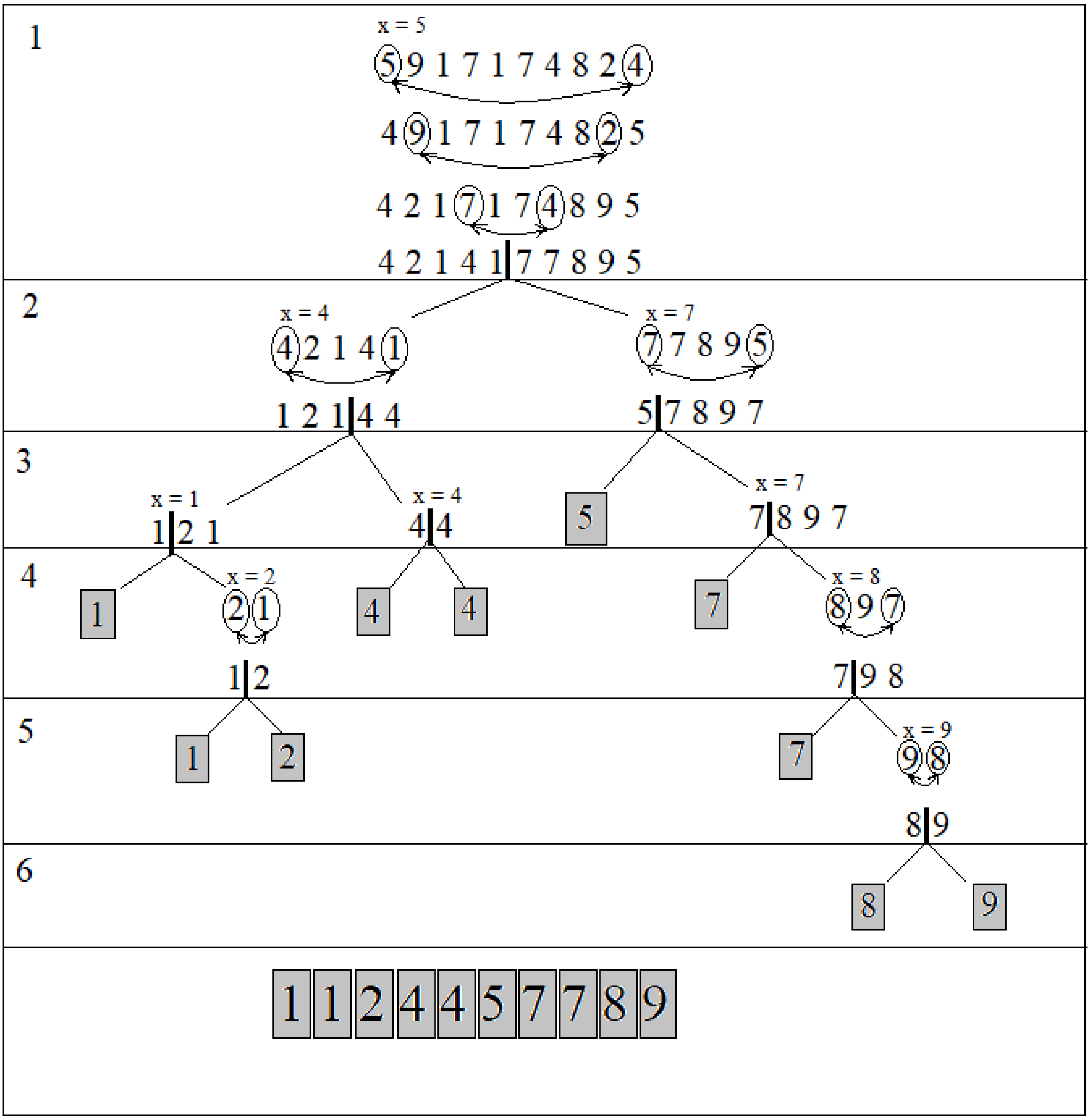
3. Злиттям


5
9
1
7
1
7
4
8
2
4
5 9
17
17
48
24
1579
1478
124478
1124457789

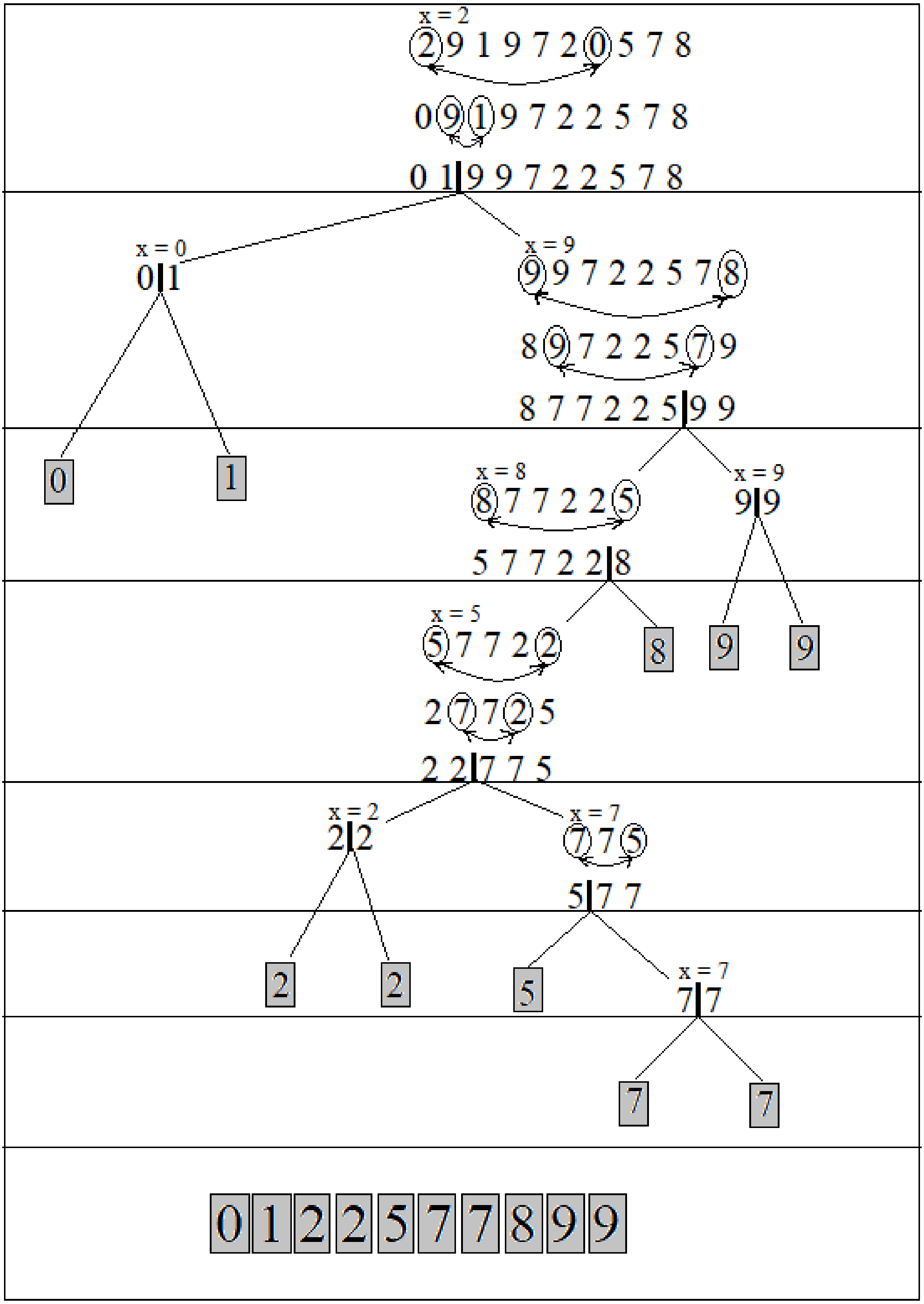

2
9
1
9
7
2
0
5
7
8
29
19
27
05
78
1299
0257
025778
0122577899

4. Швидким сортуванням

1) 5 9 1 7 1 7 4 8 2 4



2) 2 9 1 9 7 2 0 5 7 8



Завдання 6

Згенерувати дві псевдовипадкові послідовності довжиною 16 двома генераторами (початкові параметри вибрати довільно).

Розв'язання

1) Використаємо формулу лінійного конгруентного метода

$$\mathrm{X}_{\mathrm{k+1}} = (\mathrm{aX_k} + \mathrm{c}) \bmod \mathrm{m}$$

Задамо коефіцієнти для обчислення елементів послідовності a = 69069, c = 5.

Покладемо $\mathrm{m} = 2^{16}$. Виберемо довільним чином стартове число послідовності $\mathrm{x}_1 = \mathbf{24}$.

$$x_2 = (ax_1 + c) mod\ m = (69069 * 24 + 5) mod\ 2^{16} =$$
$$(1657661) mod\ 65536 = \mathbf{19261};$$
$$x_3 = (ax_2 + c) mod\ m = (69069 * 19261 + 5) mod\ 2^{16} =$$
$$(1330338009) mod\ 65536 = \mathbf{22745};$$
$$x_4 = (ax_3 + c) mod\ m = (69069 * 22745 + 5) mod\ 2^{16} =$$
$$(1570974410) mod\ 65536 = \mathbf{10954};$$
$$x_5 = (ax_4 + c) mod\ m = (69069 * 10954 + 5) mod\ 2^{16} =$$
$$(756581831) mod\ 65536 = \mathbf{34247};$$
$$x_6 = (ax_5 + c) mod\ m = (69069 * 34247 + 5) mod\ 2^{16} =$$
$$(2365406048) mod\ 65536 = \mathbf{15200};$$
$$x_7 = (ax_6 + c) mod\ m = (69069 * 15200 + 5) mod\ 2^{16} =$$
$$(1049848805) mod\ 65536 = \mathbf{27621};$$
$$x_8 = (ax_7 + c) mod\ m = (69069 * 27621 + 5) mod\ 2^{16} =$$
$$(1907754854) mod\ 65536 = \mathbf{1894;}$$
$$x_9 = (ax_8 + c) mod\ m = (69069 * 1894 + 5) mod\ 2^{16} =$$
$$(130816691) mod\ 65536 = \mathbf{6835};$$
$$x_{10} = (ax_9 + c) mod\ m = (69069 * 6835 + 5) mod\ 2^{16} =$$
$$(472086620) mod\ 65536 = \mathbf{30812};$$
$$x_{11} = (ax_{10} + c) mod\ m = (69069 * 30812 + 5) mod\ 2^{16} =$$
$$(2128154033) mod\ 65536 = \mathbf{3505};$$

$$x_{12} = (ax_{11} + c)mod\ m = \ (69069 * 3505 + 5)mod\ 2^{16} = (242086850)mod\ 65536 = \mathbf{62402;}$$

$$x_{13} = (ax_{12} + c)mod\ m = \ (69069 * 62402 + 5)mod\ 2^{16} = (4310043743)mod\ 65536 = \mathbf{3167};$$

$$x_{14} = (ax_{13} + c)mod\ m = \ (69069 * 3167 + 5)mod\ 2^{16} = (218741528)mod\ 65536 = \mathbf{47896};$$

$$x_{15} = (ax_{14} + c)mod\ m = \ (69069 * 47896 + 5)mod\ 2^{16} = (3308128829)mod\ 65536 = \mathbf{2621};$$

$$x_{16} = (ax_{15} + c)mod\ m = \ (69069 * 2621 + 5)mod\ 2^{16} = (181029854)mod\ 65536 = \mathbf{19422};$$

2) Задамо лаги (a, b) = (4, 2) і початкові елементи $(x_0, x_1, x_2, x_3) =$ (0,4; 0,1; 0,9; 0,6).

Використовуємо формулу метода Фібоначі із запізненнями

$$X_k = \begin{cases} X_{k-a} - X_{k-b}, \text{якщо } X_{k-a} \geq X_{k-b} \\ X_{k-a} - X_{k-b} + 1, \text{якщо } \ X_{k-a} < X_{k-b} \end{cases}$$

$x_4 = x_0 - \ x_2 + 1 = \ 0{,}4 - 0{,}9 + 1 = \mathbf{0,5}$;

$x_5 = x_1 - \ x_3 + 1 = \ 0{,}1 - 0{,}6 + 1 = \mathbf{0,5}$;

$x_6 = x_2 - \ x_4 = 0{,}9 - 0{,}5 = \mathbf{0,4}$;

$x_7 = x_3 - \ x_5 = 0{,}6 - 0{,}5 = \mathbf{0,1}$;

$x_8 = x_4 - x_6 = 0{,}5 - 0{,}4 = \mathbf{0,1}$;

$x_9 = x_5 - x_7 = 0{,}5 - 0{,}1 = \mathbf{0,4}$;

$x_{10} = x_6 - x_8 = 0{,}4 - 0{,}1 = \mathbf{0,3}$;

$x_{11} = x_7 - x_9 + 1 = 0{,}1 - 0{,}4 + 1 = \mathbf{0,7}$;

$x_{12} = x_8 - x_{10} + 1 = 0{,}1 - 0{,}3 + 1 = \mathbf{0,8}$;

$x_{13} = x_9 - x_{11} + 1 = 0{,}4 - 0{,}7 + 1 = \mathbf{0,7}$;

$x_{14} = x_{10} - x_{12} + 1 = 0{,}3 - 0{,}8 + 1 = \mathbf{0,5}$;

$x_{15} = x_{11} - x_{13} = 0{,}7 - 0{,}7 = \mathbf{0}$;

$x_{16} = x_{12} - x_{14} = 0{,}8 - 0{,}5 = \mathbf{0,3}$;

$x_{17} = x_{13} - x_{15} = 0{,}7 - 0 = \mathbf{0,7}$;

$$x_{18} = x_{14} - x_{16} = 0{,}5 - 0{,}3 = \mathbf{0,2};$$

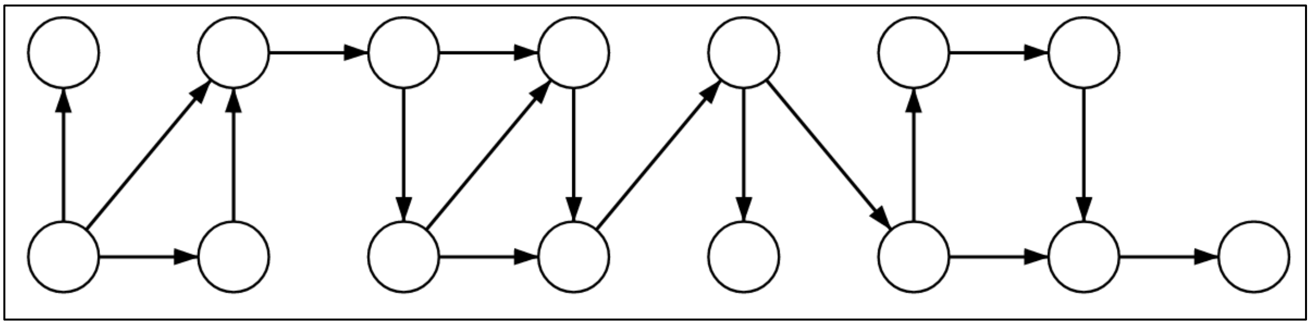
$$x_{19} = x_{15} - x_{17} + 1 = 0 - 0{,}7 + 1 = \mathbf{0,3};$$

$$x_{20} = x_{16} - x_{18} = 0{,}3 - 0{,}2 = \mathbf{0,1};$$

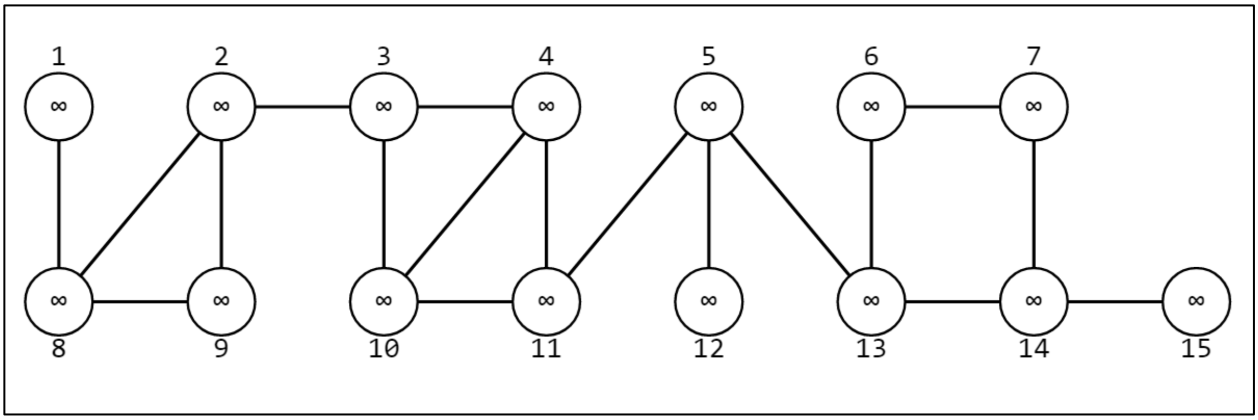
Завдання 7

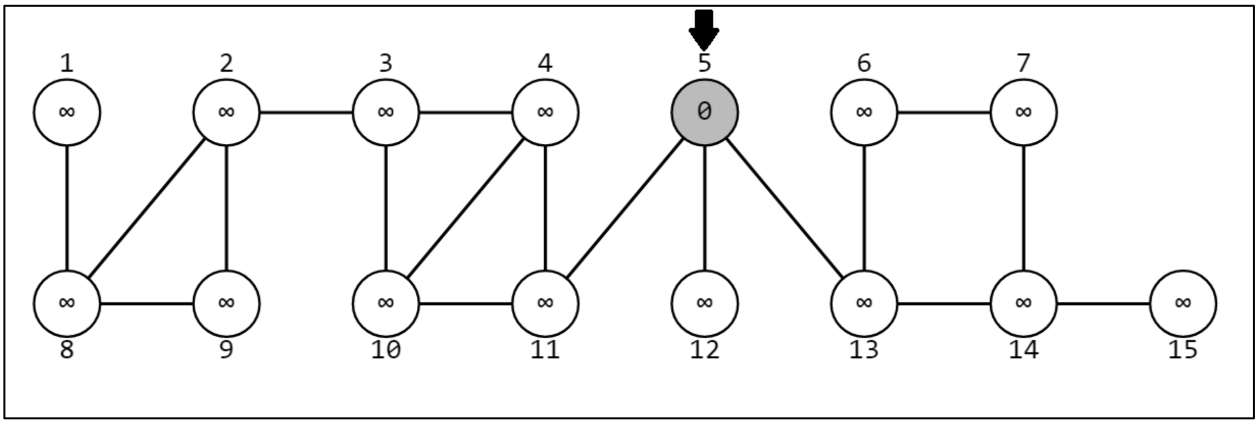
Побудувати орієнтований граф без циклів з 15-ма вершинами та 18-ма ребрами, не більше 3-х ребер при кожній вершині. Виконати пошук в глибину, ширину та провести топологічне сортування.

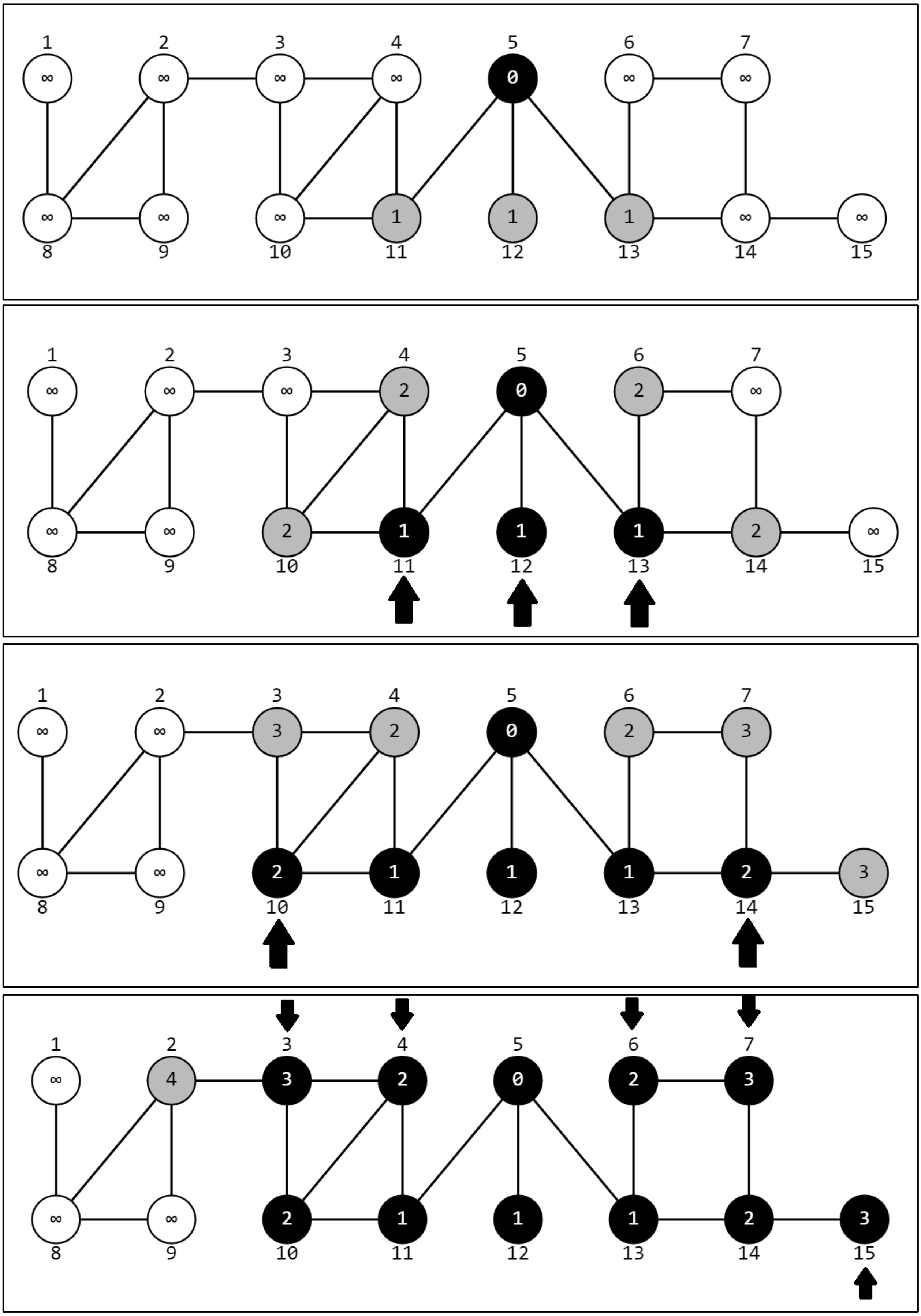
Розв'язання

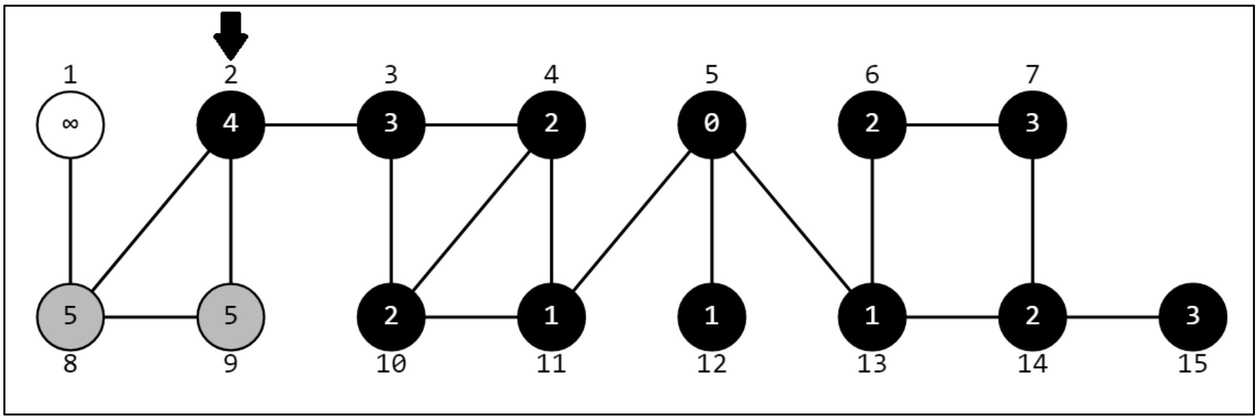


1) Пошук в ширину

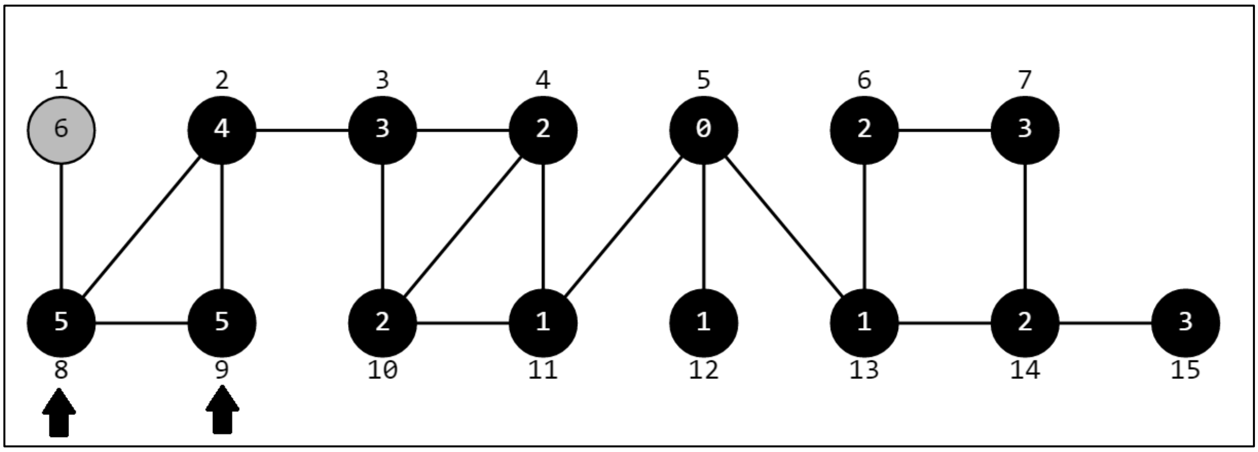




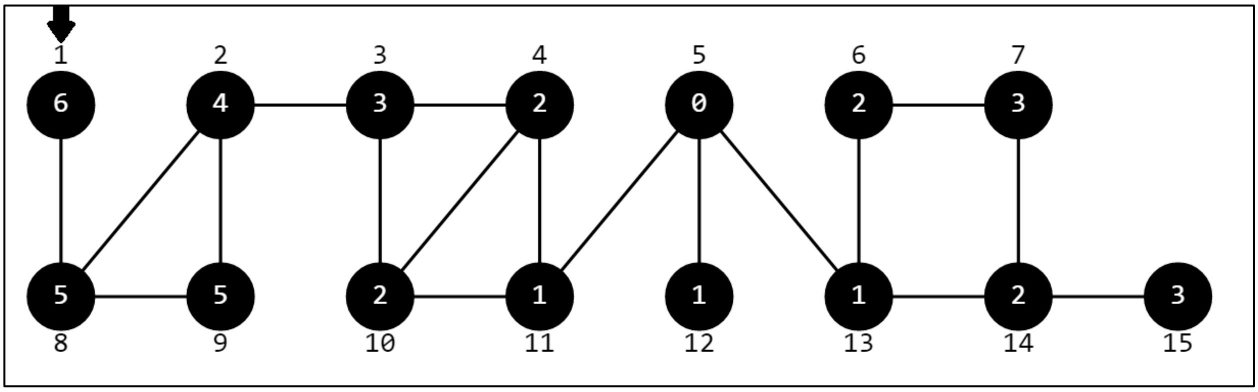

1 2 3 4 5 6 7
∞ ∞ ∞ ∞ 0 ∞ ∞
∞ ∞ ∞ 1 1 1 ∞ ∞
8 9 10 11 12 13 14 15
1 2 3 4 5 6 7
∞ ∞ ∞ 2 0 2 ∞
∞ ∞ 2 1 1 1 2 ∞
8 9 10 11 12 13 14 15
1 2 3 4 5 6 7
∞ ∞ 3 2 0 2 3
∞ ∞ 2 1 1 1 2 3
8 9 10 11 12 13 14 15
1 2 3 4 5 6 7
∞ 4 3 2 0 2 3
∞ ∞ 2 1 1 1 2 3
8 9 10 11 12 13 14 15



1 2 3 4 5 6 7
∞ 4 3 2 0 2 3
5 5 2 1 1 1 2 3
8 9 10 11 12 13 14 15

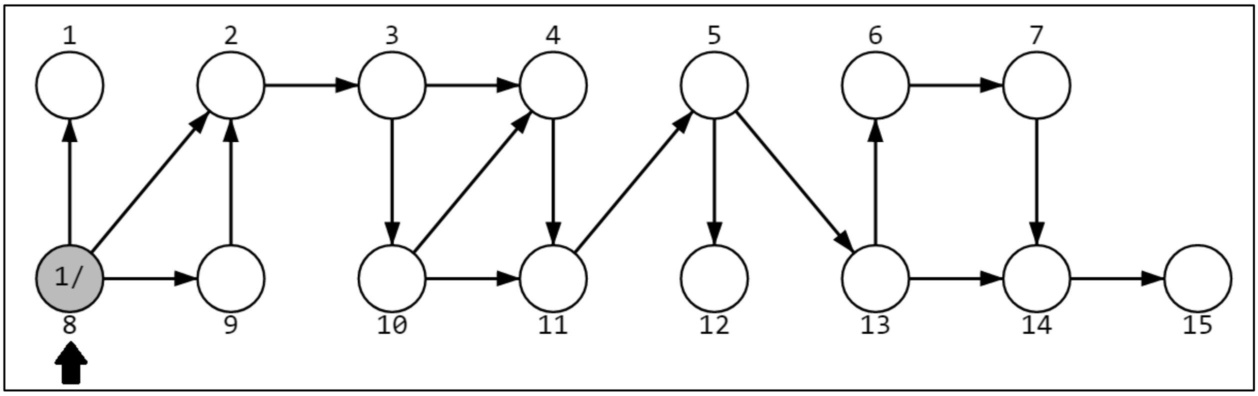


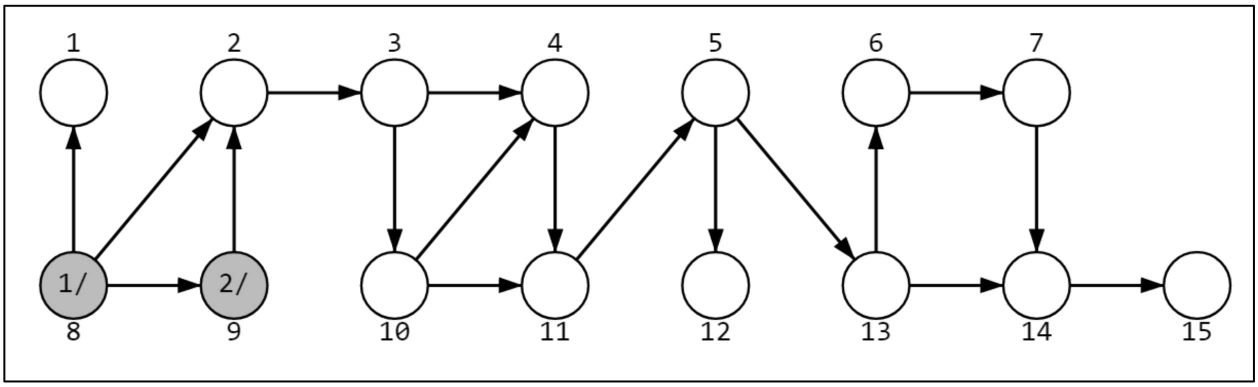
1 2 3 4 5 6 7
6 4 3 2 0 2 3
5 5 2 1 1 1 2 3
8 9 10 11 12 13 14 15

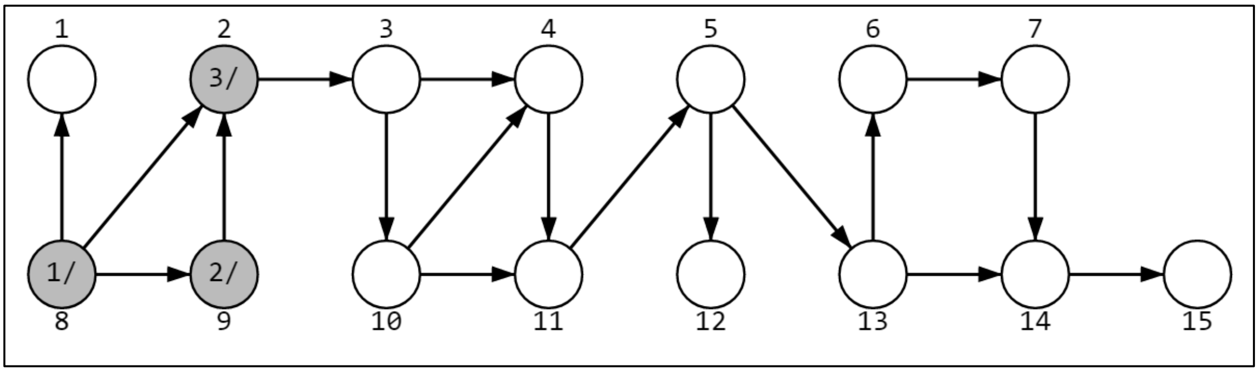


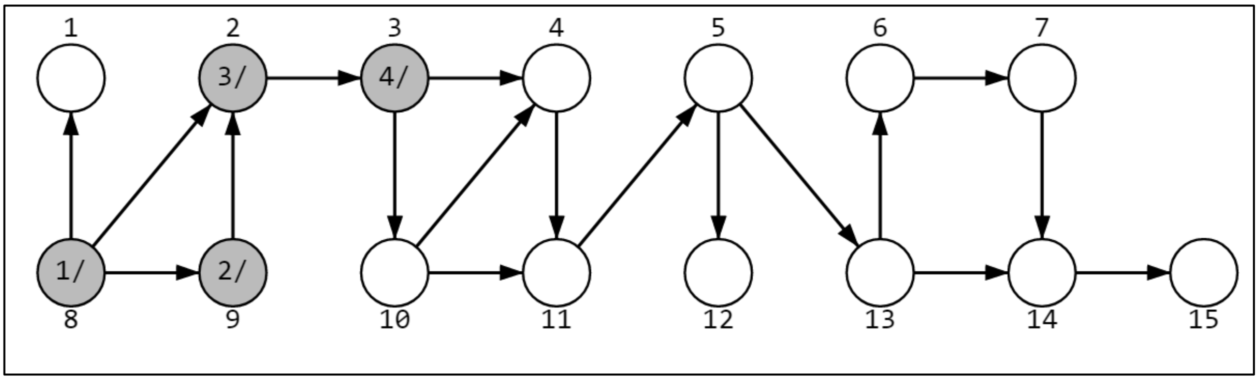
1 2 3 4 5 6 7
6 4 3 2 0 2 3
5 5 2 1 1 1 2 3
8 9 10 11 12 13 14 15

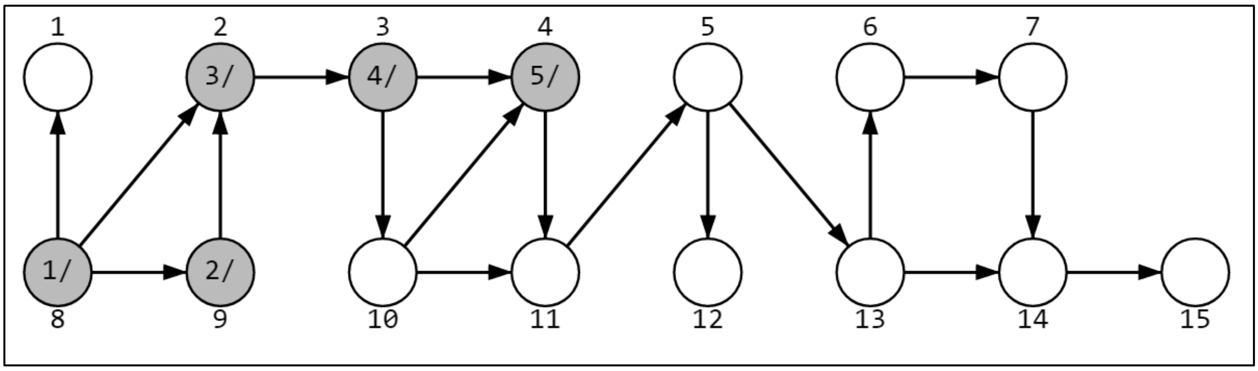
2) Пошук в глибину

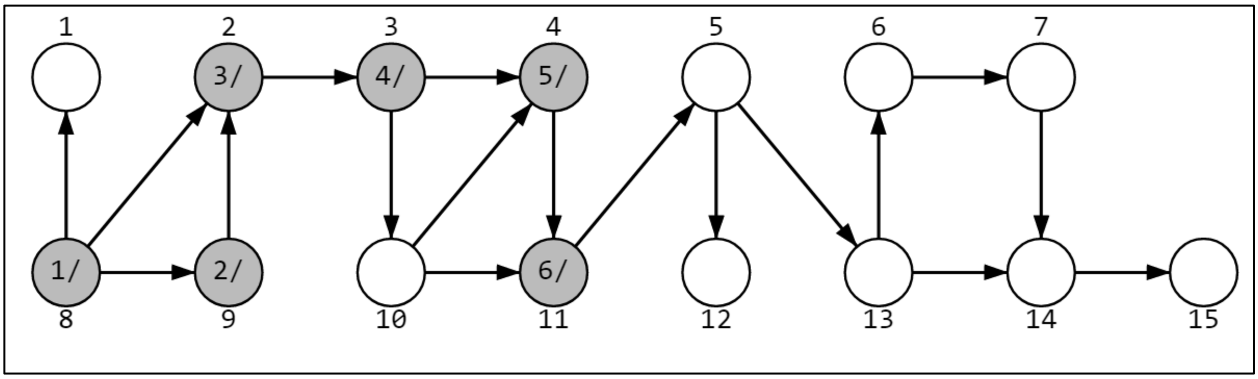


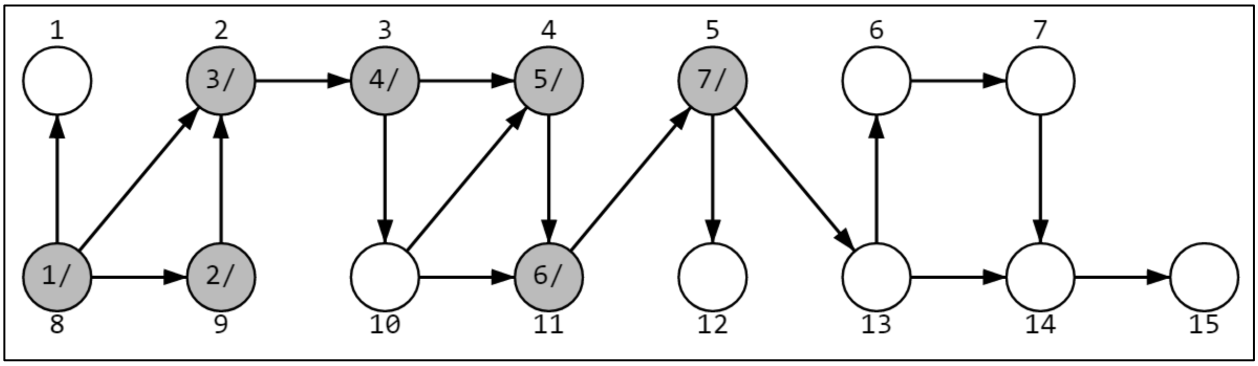


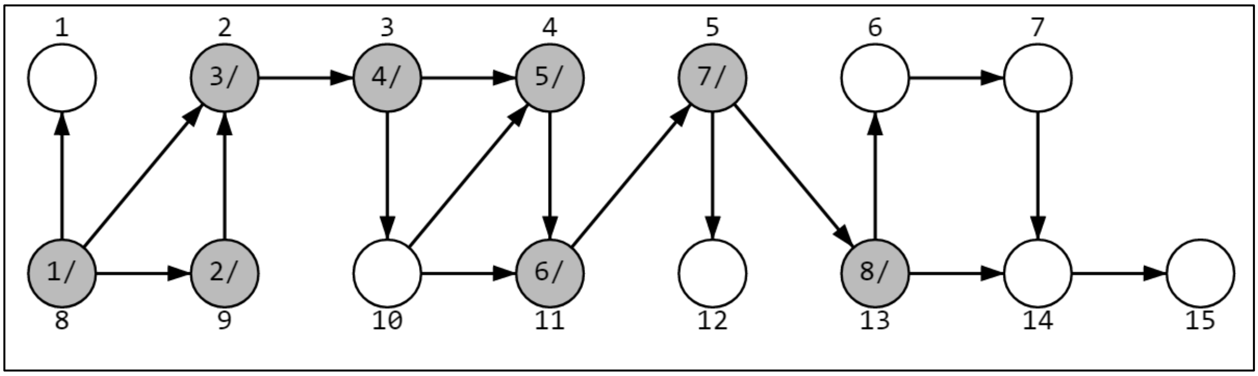


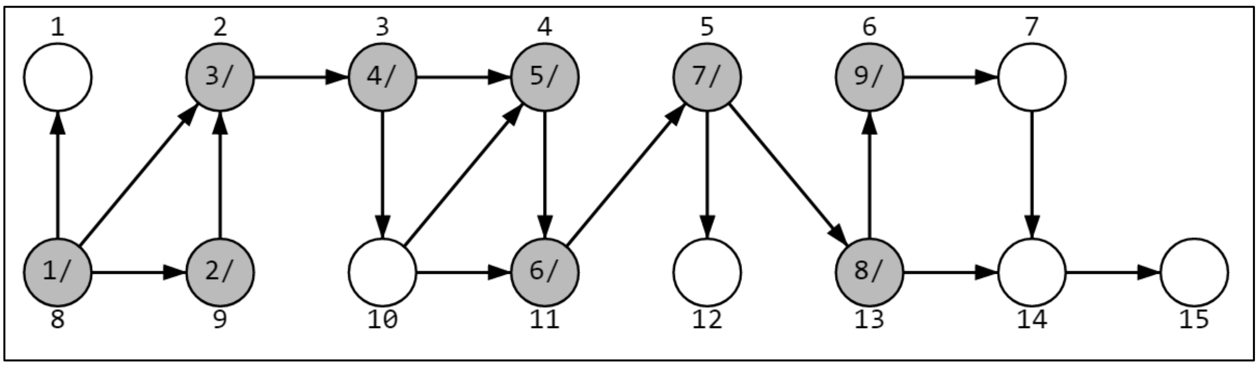


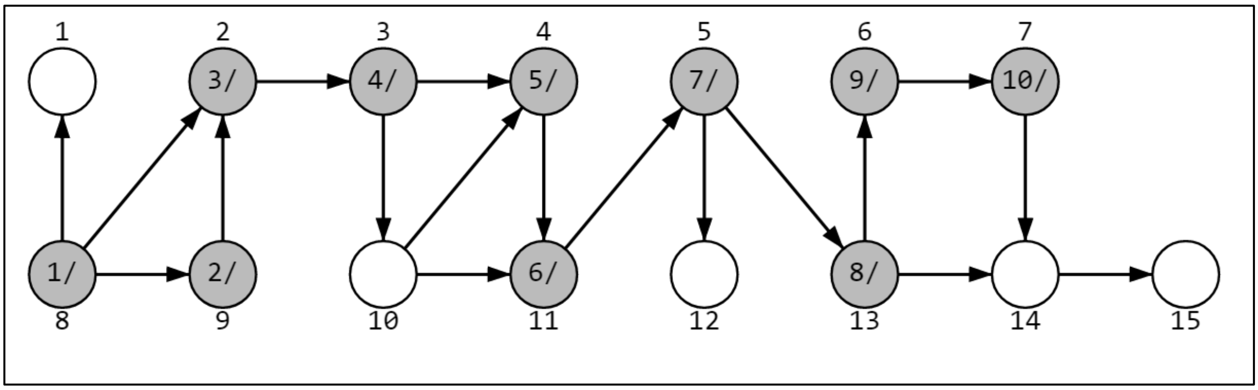

1 2 3 4 5 6 7
3/ 4/ 5/
1/ 2/
8 9 10 11 12 13 14 15

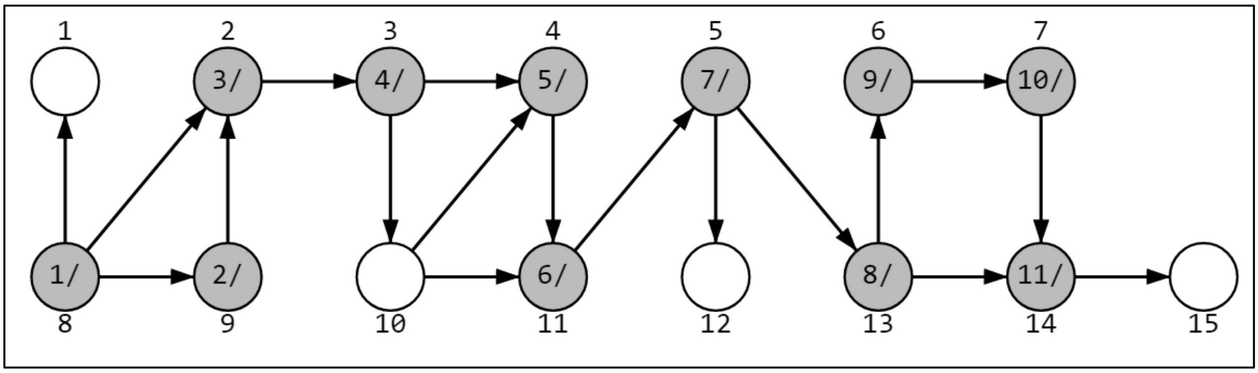

1 2 3 4 5 6 7
3/ 4/ 5/
1/ 2/ 6/
8 9 10 11 12 13 14 15

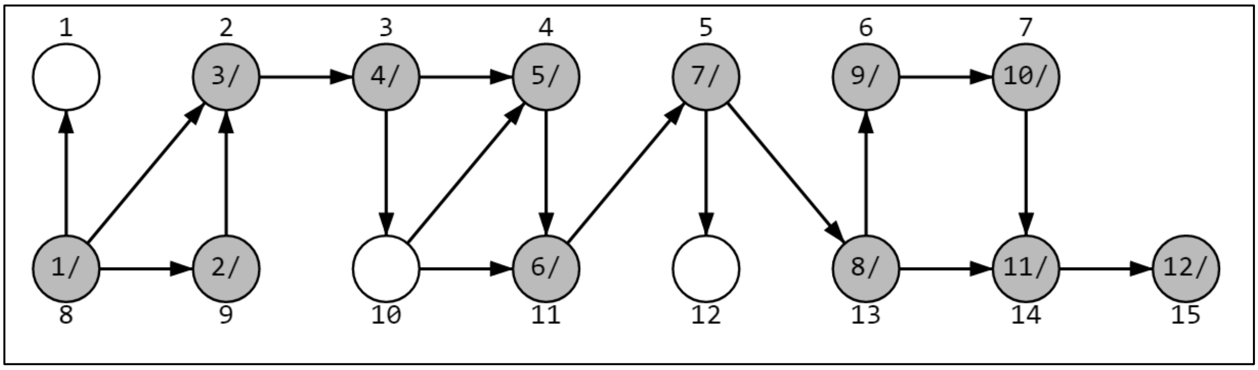

1 2 3 4 5 6 7
3/ 4/ 5/ 7/
1/ 2/ 6/
8 9 10 11 12 13 14 15

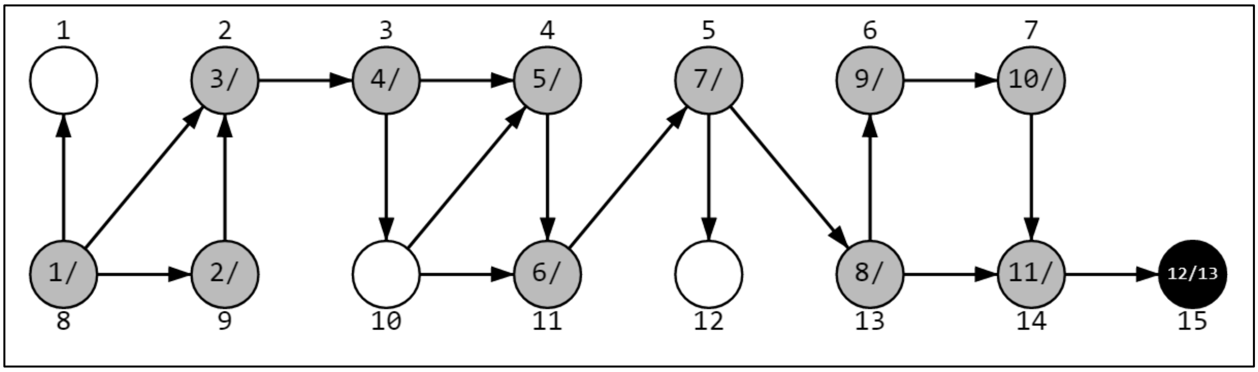

1 2 3 4 5 6 7
3/ 4/ 5/ 7/
1/ 2/ 6/ 8/
8 9 10 11 12 13 14 15

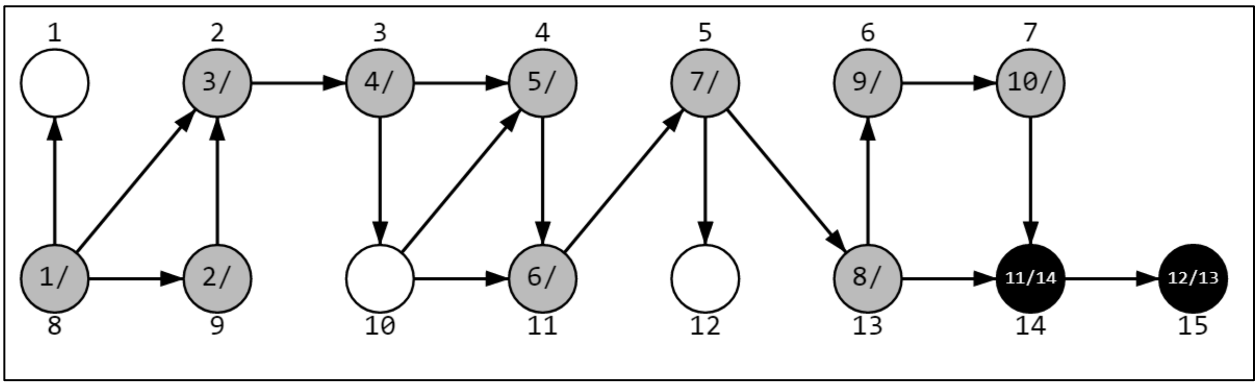

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/
1/ 2/ 6/ 8/
8 9 10 11 12 13 14 15

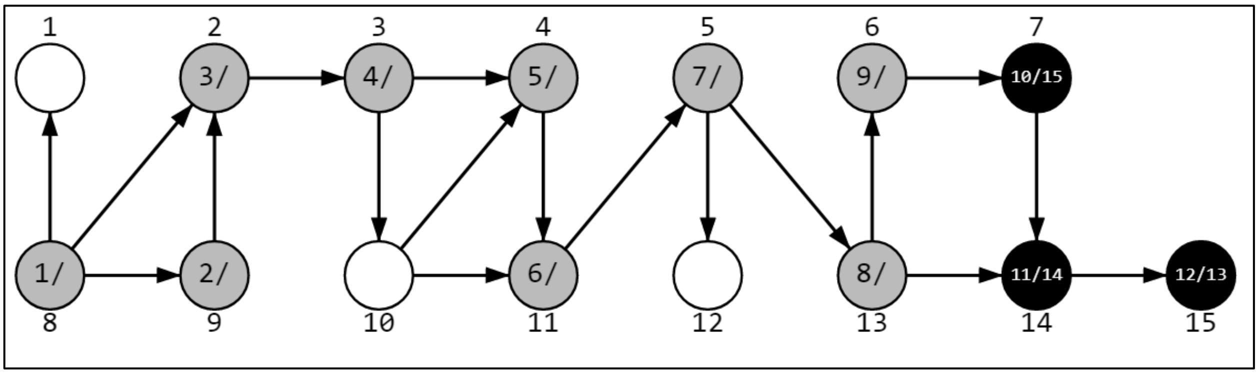

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/ 10/
1/ 2/ 6/ 8/
8 9 10 11 12 13 14 15

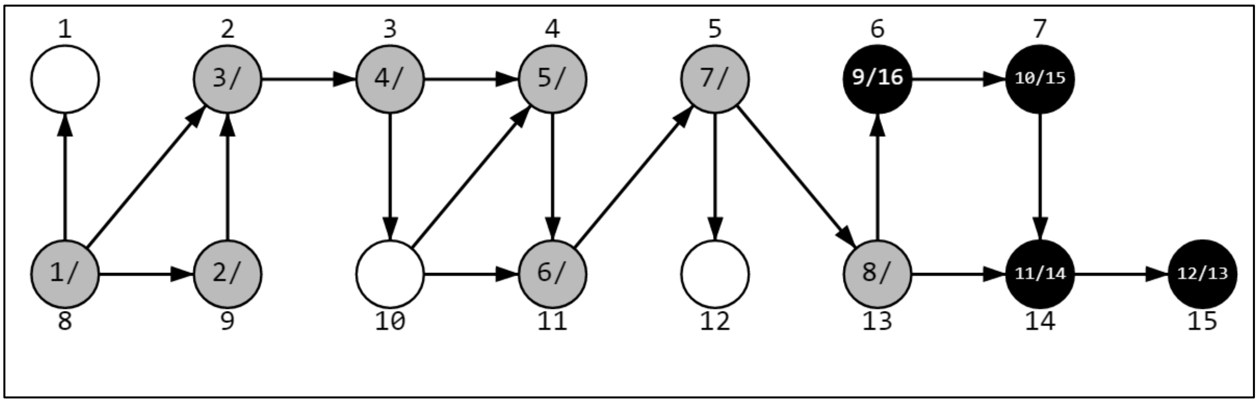

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/ 10/
1/ 2/ 6/ 8/ 11/
8 9 10 11 12 13 14 15

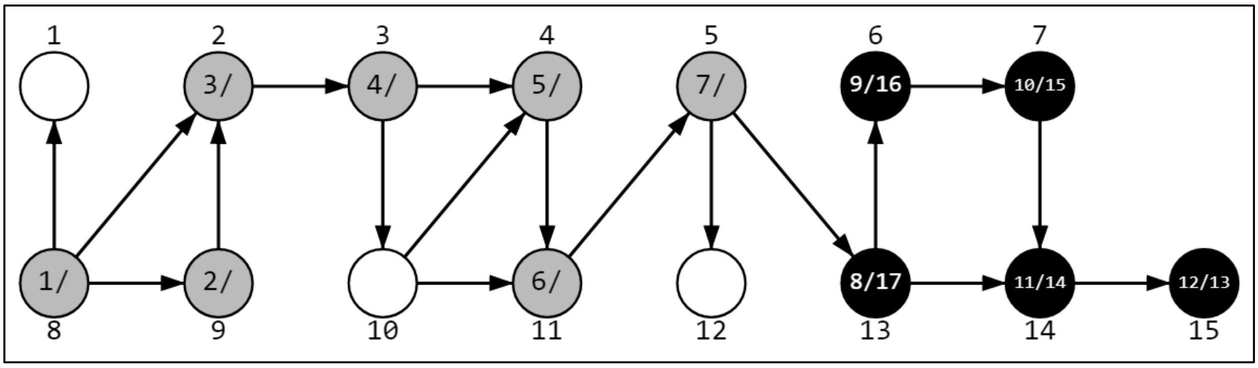

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/ 10/
1/ 2/ 6/ 8/ 11/ 12/
8 9 10 11 12 13 14 15

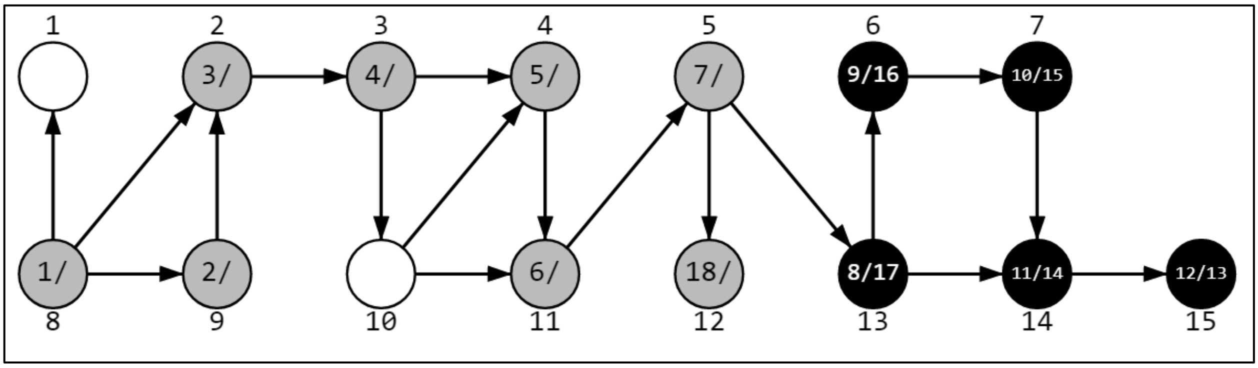

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/ 10/
1/ 2/ 6/ 8/ 11/ 12/13
8 9 10 11 12 13 14 15

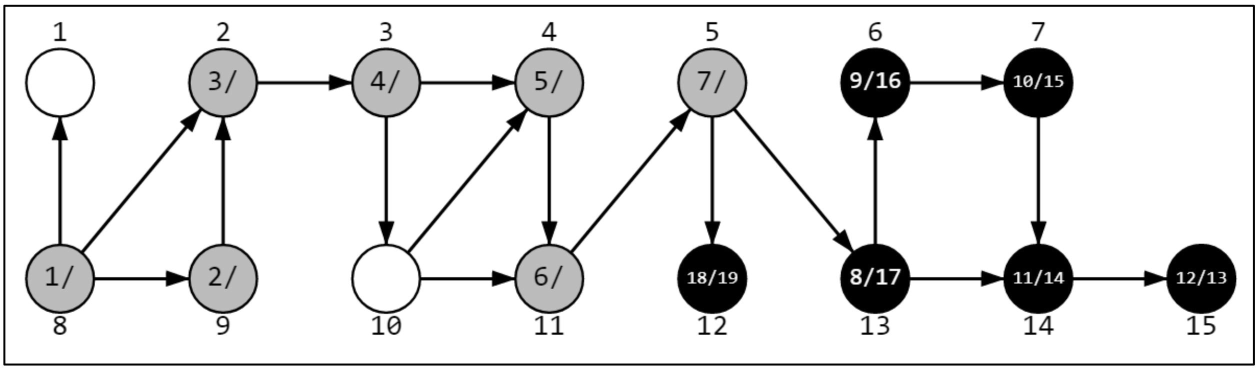

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/ 10/
1/ 2/ 6/ 8/ 11/14 12/13
8 9 10 11 12 13 14 15

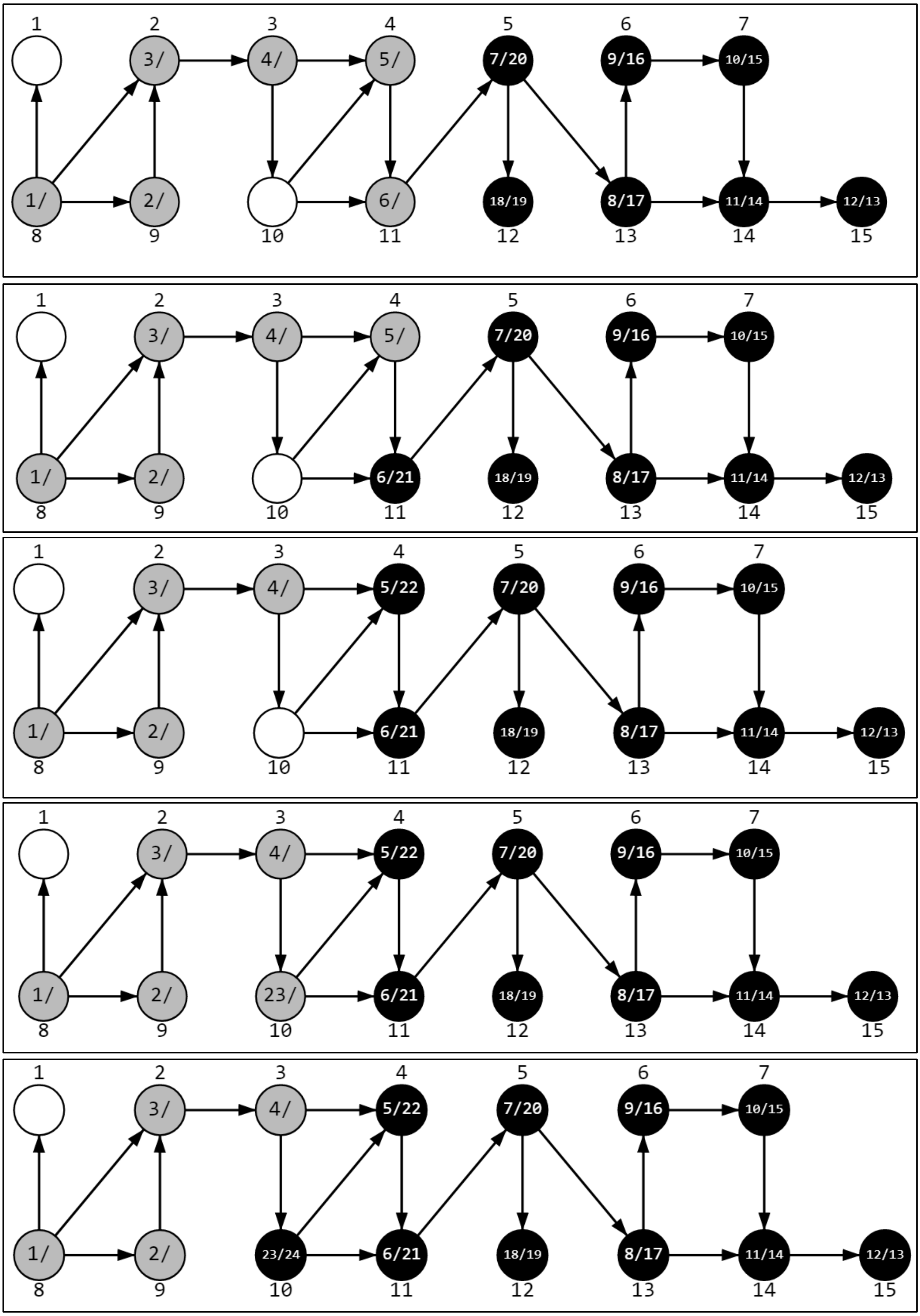

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/ 10/15
1/ 2/ 6/ 8/ 11/14 12/13
8 9 10 11 12 13 14 15

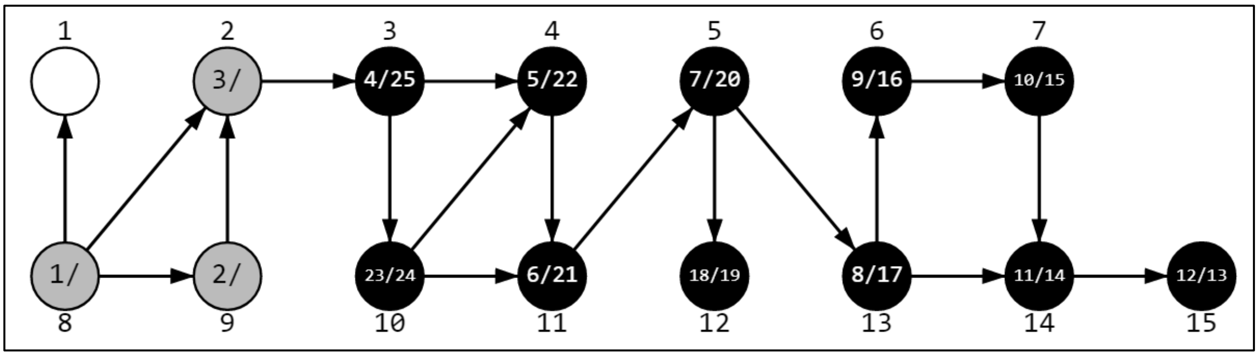

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/16 10/15
1/ 2/ 6/ 8/ 11/14 12/13
8 9 10 11 12 13 14 15

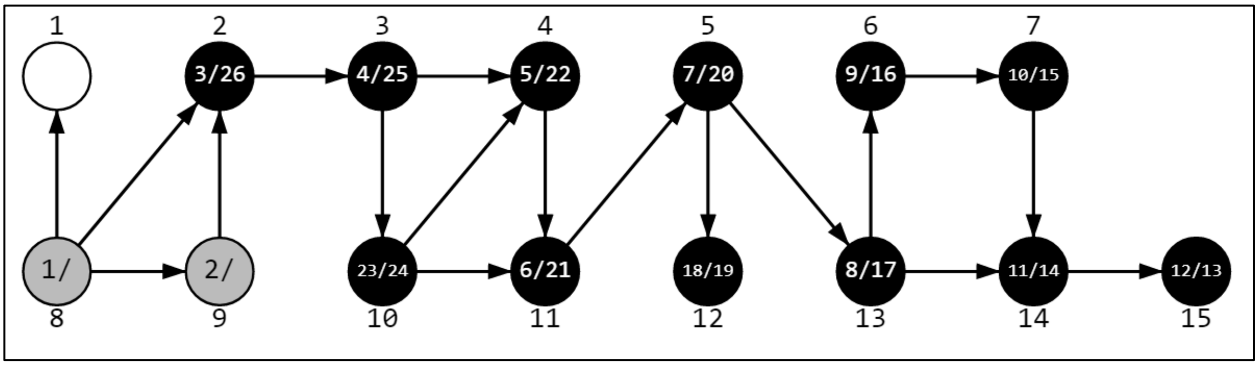

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/16 10/15
1/ 2/ 6/ 8/17 11/14 12/13
8 9 10 11 12 13 14 15

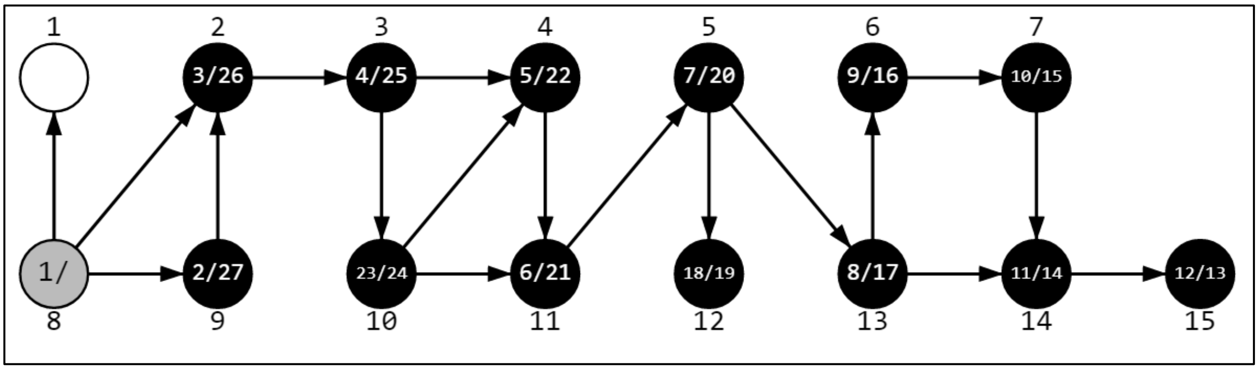

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/16 10/15
1/ 2/ 6/ 18/ 8/17 11/14 12/13
8 9 10 11 12 13 14 15

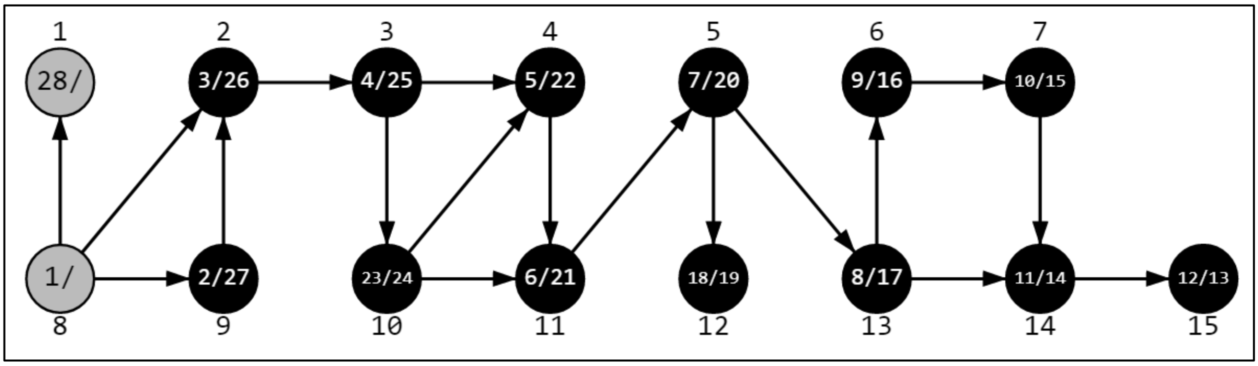

1 2 3 4 5 6 7
3/ 4/ 5/ 7/ 9/16 10/15
1/ 2/ 6/ 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15

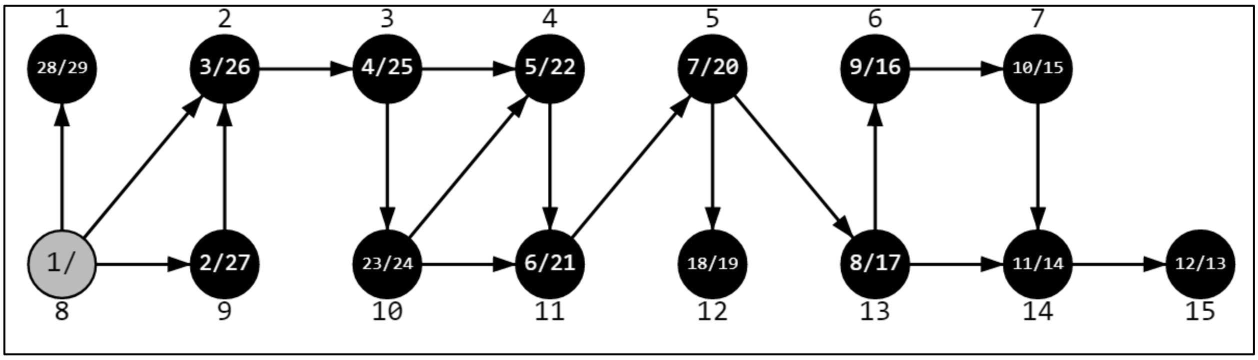

1 2 3 4 5 6 7
3/ 4/ 5/ 7/20 9/16 10/15
1/ 2/ 6/ 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15
1 2 3 4 5 6 7
3/ 4/ 5/ 7/20 9/16 10/15
1/ 2/ 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15
1 2 3 4 5 6 7
3/ 4/ 5/22 7/20 9/16 10/15
1/ 2/ 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15
1 2 3 4 5 6 7
3/ 4/ 5/22 7/20 9/16 10/15
1/ 2/ 23/ 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15
1 2 3 4 5 6 7
3/ 4/ 5/22 7/20 9/16 10/15
1/ 2/ 23/24 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15

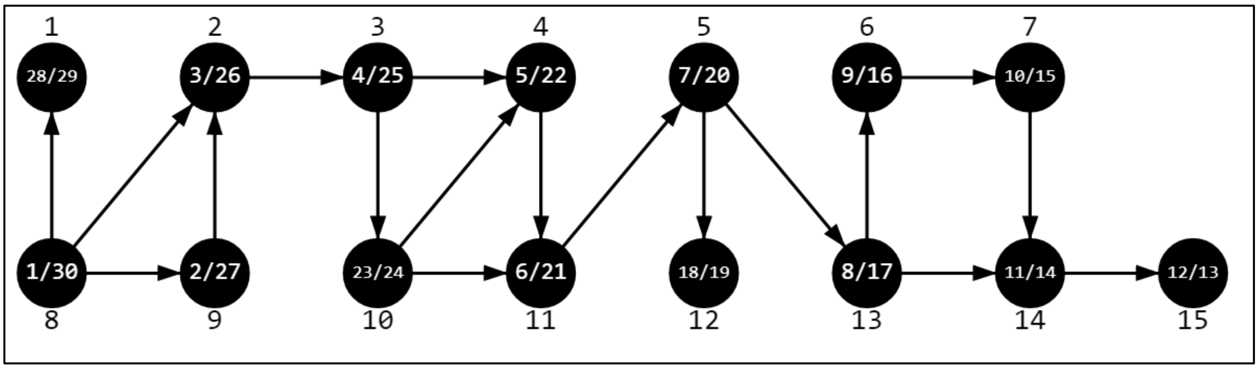

1 2 3 4 5 6 7
3/ 4/25 5/22 7/20 9/16 10/15
1/ 2/ 23/24 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15


1 2 3 4 5 6 7
3/26 4/25 5/22 7/20 9/16 10/15
1/ 2/ 23/24 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15

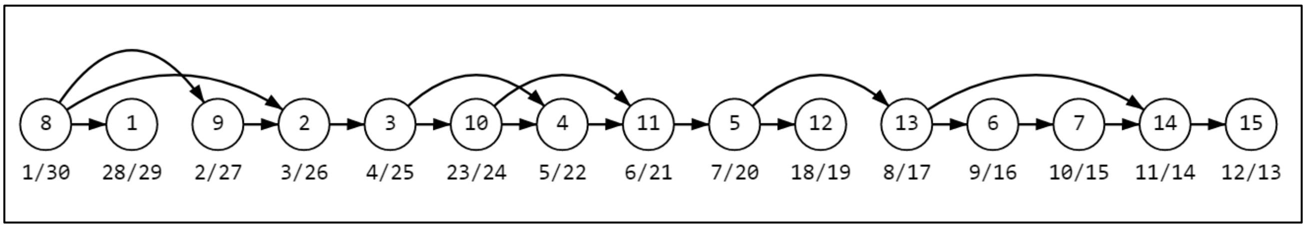

1 2 3 4 5 6 7
3/26 4/25 5/22 7/20 9/16 10/15
1/ 2/27 23/24 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15


1 2 3 4 5 6 7
28/ 3/26 4/25 5/22 7/20 9/16 10/15
1/ 2/27 23/24 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15


1 2 3 4 5 6 7
28/29 3/26 4/25 5/22 7/20 9/16 10/15
1/ 2/27 23/24 6/21 18/19 8/17 11/14 12/13
8 9 10 11 12 13 14 15



3) Топологічне сортування



Завдання 8

Взяти будь-яку послідовність із завдання 6 та використати її числа, як координати $x_1, y_1;\ x_2, y_2; \ldots\ x_8, y_8$. Знайти опуклу оболонку цієї множини точок.

Розв'язання №1

$q_1 = x_1 y_1 = (0{,}5; 0{,}4)$

$q_2 = x_2 y_2 = (0{,}1; 0{,}1)$

$q_3 = x_3 y_3 = (0{,}4; 0{,}3)$

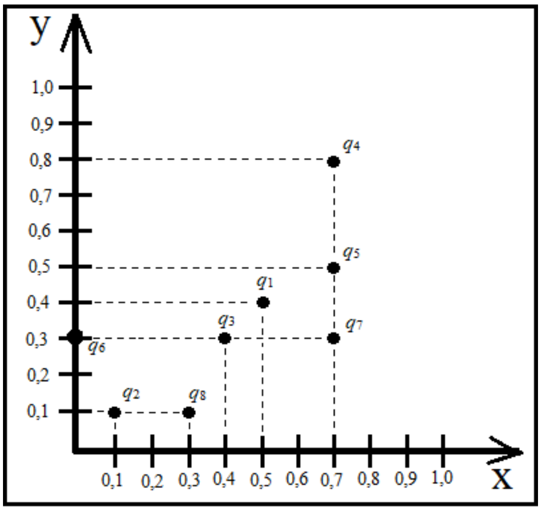
$q_4 = x_4 y_4 = (0{,}7; 0{,}8)$

$q_5 = x_5 y_5 = (0{,}7; 0{,}5)$

$q_6 = x_6 y_6 = (0; 0{,}3)$

$q_7 = x_7 y_7 = (0{,}7; 0{,}3)$

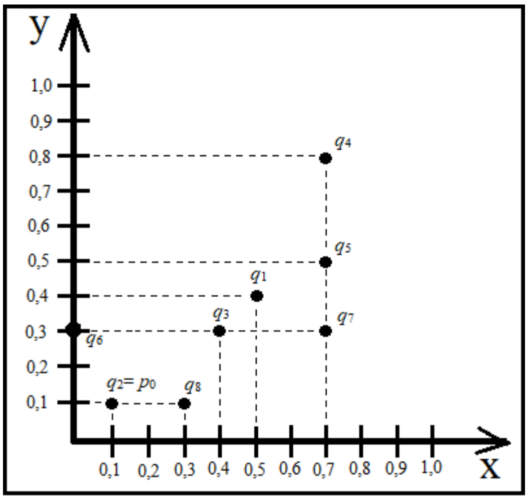
$q_8 = x_8 y_8 = (0{,}3; 0{,}1)$



Для побудови опуклої оболонки будемо використовувати алгоритм обходу Джарвіса, для порівняння точок будемо використовувати знак векторного добутку $(p_{best} - p_{i-1}) \times (q_j - p_{i-1})$. Будемо позначати як p_i точку, яку на i-му кроці алгоритму було включено до опуклої оболонки Q.

Крок 0. Знаходимо початкову точку в обході. Це буде нижня ліва точка, в даному випадку $q_2 = (0{,}1;\ 0{,}1)$. Покладемо $i = 0$, $p_0 = q_2 = (0{,}1;\ 0{,}1)$.

Включаємо точку до оболонки: $Q = \{q_2\}$.



Крок 1. Знаходимо наступну за q_3 точку на опуклій оболонці.

Покладемо $\boldsymbol{i} = \mathbf{1}$. В якості претендента на включення до опуклої оболонки беремо першу точку не з оболонки і позначаємо її p_{best}. В даному випадку $\boldsymbol{p_{best}} = \boldsymbol{q_1}\mathbf{(0,5;\ 0,4)}$, тоді порівняння починаємо з другої точки.

Покладемо $j = 2$. Так як точка q_2 – це єдина точка в оболонці, переходимо до наступної.

Покладемо $j = j + 1 = 3$. Порівняємо $q_j = q_3(0,4;\ 0,3)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_0) \times (q_3 - p_0) = (0{,}4; 0{,}3) \times (0{,}3; 0{,}2) = \det\begin{pmatrix} 0{,}4 & 0{,}3 \\ 0{,}3 & 0{,}2 \end{pmatrix} = 0{,}08 - 0{,}09 = -0{,}01 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_3(0,4;\ 0,3)$.

Покладемо $j = j + 1 = 4$. Порівняємо $q_j = q_4(0,7;\ 0,8)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_0) \times (q_4 - p_0) = (0{,}3; 0{,}2) \times (0{,}6; 0{,}7) = \det\begin{pmatrix} 0{,}3 & 0{,}6 \\ 0{,}2 & 0{,}7 \end{pmatrix} = 0{,}21 - 0{,}12 = 0{,}9 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 5$. Порівняємо $q_j = q_5(0,7;\ 0,5)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_0) \times (q_5 - p_0) = (0{,}3; 0{,}2) \times (0{,}6; 0{,}4) = \det\begin{pmatrix} 0{,}3 & 0{,}6 \\ 0{,}2 & 0{,}4 \end{pmatrix} = 0{,}12 - 0{,}12 = 0.$

Так як векторний добуток дорівнює нулю, то обираємо ту точку, яка розташована далі від точки p_0. В нашому випадку це точка q_5, $p_{best} = q_5(0,7;\ 0,5)$.

Покладемо $j = j + 1 = 6$. Порівняємо $q_j = q_6(0;\ 0,3)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_0) \times (q_6 - p_0) = (0{,}6; 0{,}4) \times (-0{,}1; 0{,}2) = \det\begin{pmatrix} 0{,}6 & -0{,}1 \\ 0{,}4 & 0{,}2 \end{pmatrix} = 0{,}12 + 0{,}04 = 0{,}16 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 7$. Порівняємо $q_j = q_7(0,7;\ 0,3)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_0) \times (q_7 - p_0) = (0{,}6; 0{,}4) \times (0{,}6; 0{,}2) = \det\begin{pmatrix} 0{,}6 & 0{,}6 \\ 0{,}4 & 0{,}2 \end{pmatrix} = 0{,}12 - 0{,}24 = -0{,}12 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_7(0,7;\ 0,3)$.

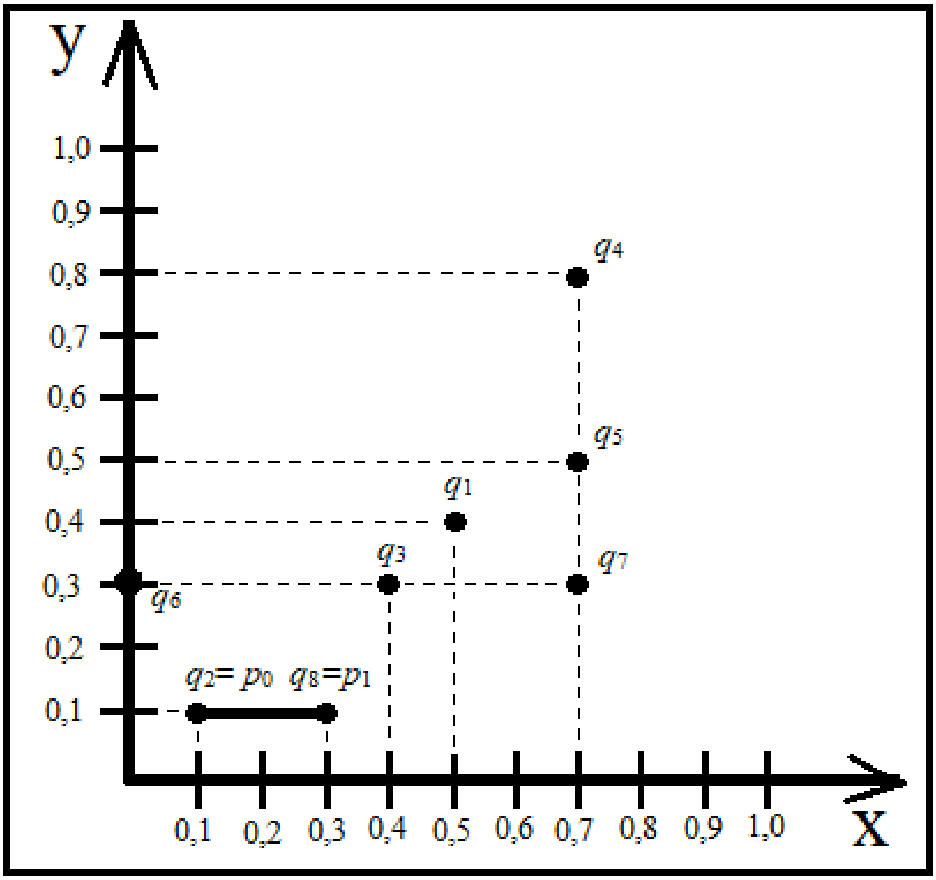
Покладемо $j = j + 1 = 8$. Порівняємо $q_j = q_8(0,3;\ 0,1)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_0) \times (q_8 - p_0) = (0{,}6; 0{,}2) \times (0{,}2; 0) = \det\begin{pmatrix} 0{,}6 & 0{,}2 \\ 0{,}3 & 0 \end{pmatrix} = 0 - 0{,}06 = -0{,}06 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_8(0,3;\ 0,1)$.

Усі точки розглянуто, отже, $p_1 = q_8(0,3;\ 0,1)$.

Включаємо цю точку в оболонку: $Q = \{q_2, q_8\}$.



Крок 2. Знаходимо наступну за q_8 точку на опуклій оболонці.

Покладемо $\boldsymbol{i}$ **= 2**. В якості претендента візьмемо $\boldsymbol{p_{best}} = \boldsymbol{q_1}$**(0,5; 0,4)**, тоді порівняння починаємо з другої точки.

Покладемо $j = 2$. Порівняємо $q_j = q_2(0,1; 0,1)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_1) \times (q_2 - p_1) = (0,2; 0,3) \times (-0,2; 0) = \det\begin{pmatrix} 0,2 & -0,2 \\ 0,3 & 0 \end{pmatrix} = 0 + 0,06 = 0,06 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 3$. Порівняємо $q_j = q_3(0,4; 0,3)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_1) \times (q_3 - p_1) = (0,2; 0,3) \times (0,1; 0,2) = \det\begin{pmatrix} 0,2 & 0,1 \\ 0,3 & 0,2 \end{pmatrix} = 0,04 - 0,03 = 0,01 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 4$. Порівняємо $q_j = q_4(0,7; 0,8)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_1) \times (q_4 - p_1) = (0,2; 0,3) \times (0,4; 0,7) = \det\begin{pmatrix} 0,2 & 0,4 \\ 0,3 & 0,7 \end{pmatrix} = 0,14 - 0,12 = 0,02 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 5$. Порівняємо $q_j = q_5(0,7; 0,5)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_1) \times (q_5 - p_1) = (0,2; 0,3) \times (0,4; 0,4) = \det\begin{pmatrix} 0,2 & 0,4 \\ 0,3 & 0,4 \end{pmatrix} = 0,08 - 0,12 = -0,04 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_5(0,7; 0,5)$.

Покладемо $j = j + 1 = 6$. Порівняємо $q_j = q_6(0; 0,3)$ з p_{best}.

Запишемо векторний добуток $(p_{best} - p_1) \times (q_6 - p_1) = (0,4; 0,4) \times (-0,3; 0,2) = \det\begin{pmatrix} 0,4 & -0,3 \\ 0,4 & 0,2 \end{pmatrix} = 0,08 + 0,12 = 0,20 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 7$. Порівняємо $q_j = q_7(0,7;\ 0,3)$ з p_{best}.

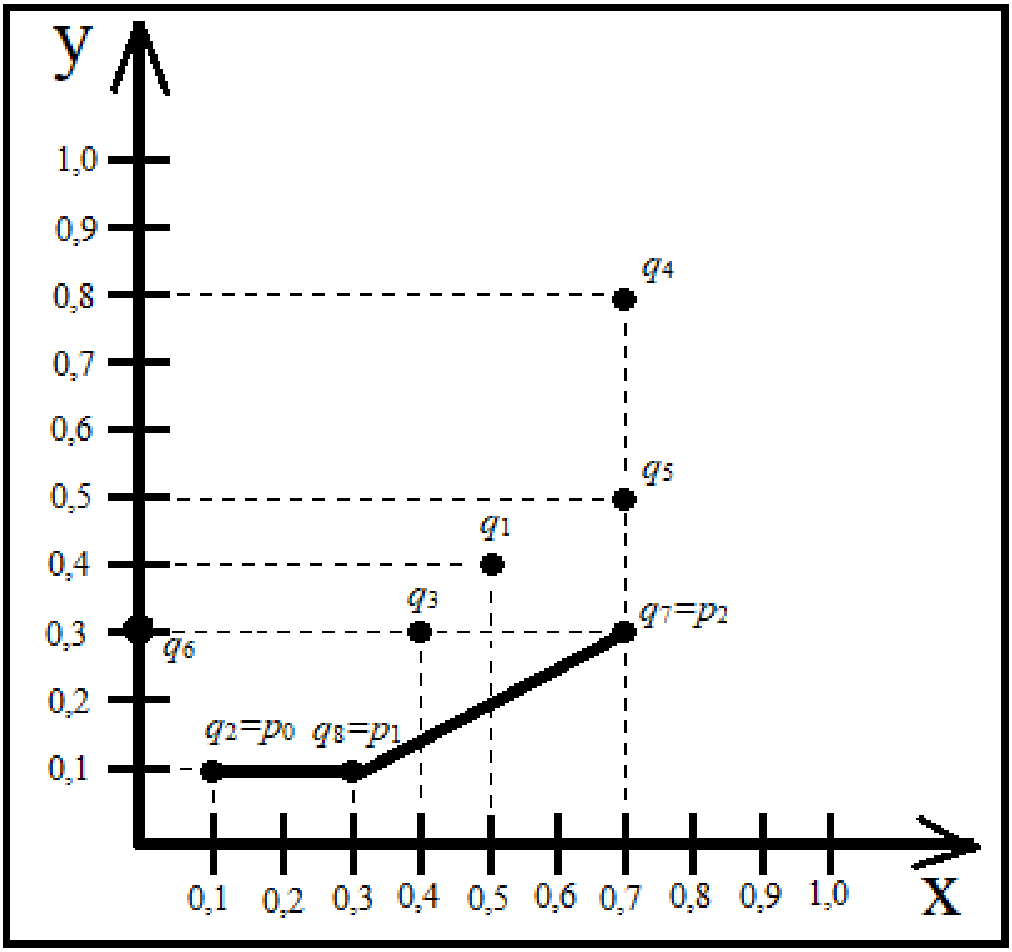
Запишемо векторний добуток $(p_{best} - p_1) \times (q_7 - p_1) = (0{,}4; 0{,}4) \times (0{,}4; 0{,}2) = \det\begin{pmatrix} 0{,}4 & 0{,}4 \\ 0{,}4 & 0{,}2 \end{pmatrix} = 0{,}08 - 0{,}16 = -0{,}08 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_7(0,7;\ 0,3)$.

Покладемо $j = j + 1 = 8$. Так як точка q_8 уже включена до опуклої оболонки, то переходимо до наступної точки.

Усі точки розглянуто, отже, $p_2 = q_7(0,7;\ 0,3)$.

Включаємо цю точку в оболонку: $Q = \{q_2, q_8, q_7\}$.



Крок 3. Знаходимо наступну за q_7 точку на опуклій оболонці.

Покладемо $\boldsymbol{i} = \mathbf{3}$. В якості претендента візьмемо $\boldsymbol{p_{best}} = \boldsymbol{q_1}\mathbf{(0,5;\ 0,4)}$, тоді порівняння починаємо з другої точки.

Покладемо $j = 2$. Порівняємо $q_j = q_2(0,1; 0,1)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_2) \times (q_2 - p_2) = (-0,2; 0,1) \times (-0,6; -0,2) = \det\begin{pmatrix} -0,2 & -0,6 \\ 0,1 & -0,2 \end{pmatrix} = 0,04 + 0,06 = 0,10 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 3$. Порівняємо $q_j = q_3(0,4; 0,3)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_2) \times (q_3 - p_2) = (-0,2; 0,1) \times (-0,3; 0) = \det\begin{pmatrix} -0,2 & -0,3 \\ 0,1 & 0 \end{pmatrix} = 0 + 0,03 = 0,03 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 4$. Порівняємо $q_j = q_4(0,7; 0,8)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_2) \times (q_4 - p_2) = (-0,2; 0,1) \times (0; 0,5) = \det\begin{pmatrix} -0,2 & 0 \\ 0,1 & 0,5 \end{pmatrix} = -0,10 - 0 = -0,10 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_4(0,7; 0,8)$.

Покладемо $j = j + 1 = 5$. Порівняємо $q_j = q_5(0,7; 0,5)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_2) \times (q_5 - p_2) = (0;\ 0,5) \times (0;\ 0,2) = \det\begin{pmatrix} 0 & 0 \\ 0,5 & 0,2 \end{pmatrix} = 0 - 0 = 0.$

Так як векторний добуток дорівнює нулю, то точки p_{best}, q_5 та p_2 лежать на одній прямій, обираємо ту точку, яка розташована далі від точки p_2. В нашому випадку це точка q_4, $p_{best} = q_4(0,7; 0,8)$.

Покладемо $j = j + 1 = 6$. Порівняємо $q_j = q_6(0; 0,3)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_2) \times (q_6 - p_2) = (0; 0{,}5) \times (-0{,}7; -0{,}5) = \det\begin{pmatrix} 0 & -0{,}7 \\ 0{,}5 & -0{,}5 \end{pmatrix} = 0 + 0{,}35 = 0{,}35 > 0.$

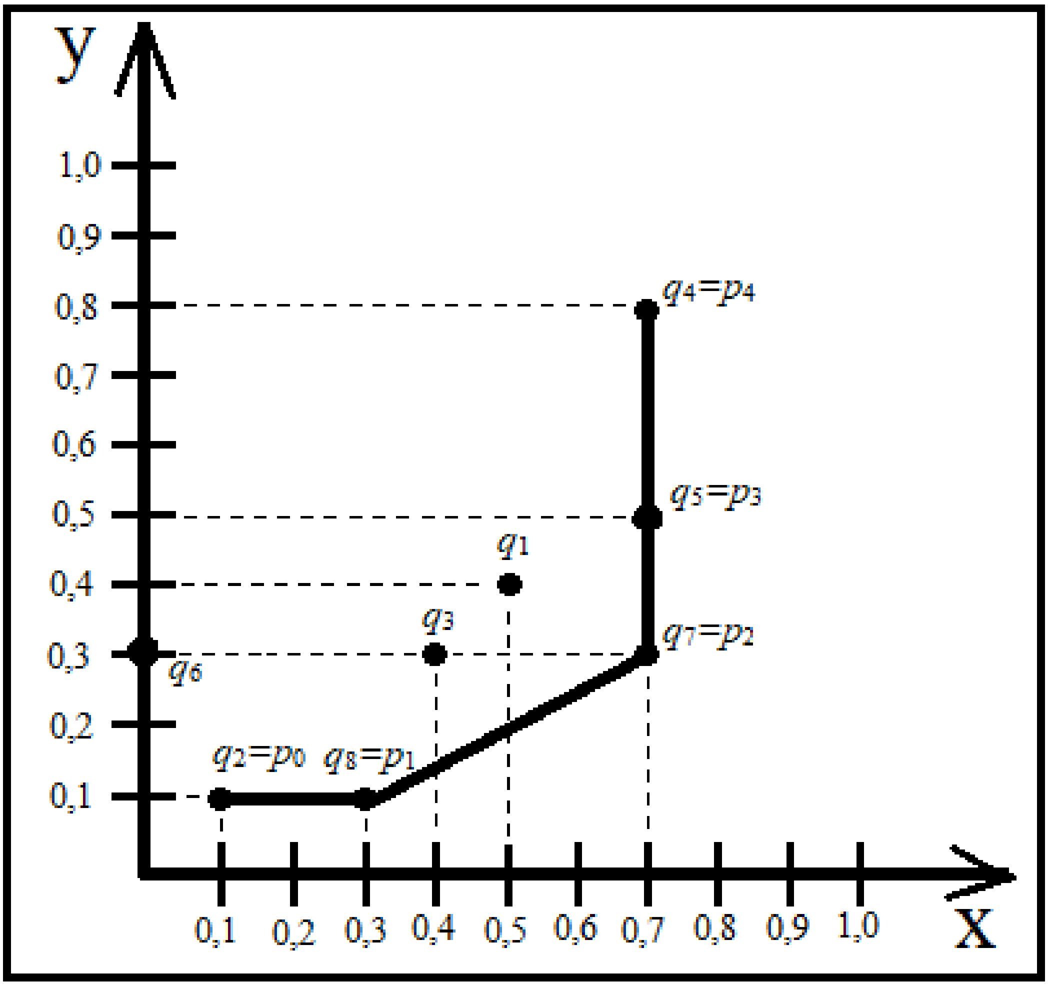
Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 7$. Так як точка q_7 уже включена до опуклої оболонки, то переходимо до наступної точки.

Покладемо $j = j + 1 = 8$. Так як точка q_8 уже включена до опуклої оболонки, то переходимо до наступної точки.

Усі точки розглянуто. Так як $p_{best} = q_4(0{,}7; 0{,}8)$, а точки p_{best}, q_5 та p_2 лежать на одній прямій, то $p_3 = q_5(0{,}7; 0{,}5)$, $p_4 = q_4(0{,}7; 0{,}8)$

Включаємо ці точки в оболонку: $Q = \{q_2, q_8, q_7, q_5, q_4\}$.



Так як на минулому кроці ми додали до оболонки одразу дві точки, то вважаємо, що ми зробили два кроки. Тому наступний крок - під номером 5.

Крок 5. Знаходимо наступну за q_4 точку на опуклій оболонці.

Покладемо $\boldsymbol{i} = \mathbf{5}$. В якості претендента візьмемо $\boldsymbol{p_{best}} = \boldsymbol{q_1}\mathbf{(0{,}5;\ 0{,}4)}$, тоді порівняння починаємо з другої точки.

Покладемо $j = 2$. Порівняємо $q_j = q_2(0{,}1;\ 0{,}1)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_4) \times (q_2 - p_4) = (-0{,}2; -0{,}4) \times (-0{,}6; -0{,}7) = \det\begin{pmatrix}-0{,}2 & -0{,}6 \\ -0{,}4 & -0{,}7\end{pmatrix} = 0{,}14 - 0{,}24 = -0{,}10 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_2(0{,}1;\ 0{,}1)$.

Покладемо $j = j + 1 = 3$. Порівняємо $q_j = q_3(0{,}4;\ 0{,}3)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_4) \times (q_3 - p_4) = (-0{,}6; -0{,}7) \times (-0{,}3; -0{,}5) = \det\begin{pmatrix}-0{,}6 & -0{,}3 \\ -0{,}7 & -0{,}5\end{pmatrix} = 0{,}30 - 0{,}21 = 0{,}09 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 4$. Так як точка q_4 уже включена до опуклої оболонки, то переходимо до наступної точки.

Покладемо $j = j + 1 = 5$. Так як точка q_5 уже включена до опуклої оболонки, то переходимо до наступної точки.

Покладемо $j = j + 1 = 6$. Порівняємо $q_j = q_6(0;\ 0{,}3)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_4) \times (q_6 - p_4) = (-0{,}6; -0{,}7) \times (-0{,}7; -0{,}5) = \det\begin{pmatrix}-0{,}6 & -0{,}7 \\ -0{,}7 & -0{,}5\end{pmatrix} = 0{,}30 - 0{,}49 = -0{,}19 < 0.$

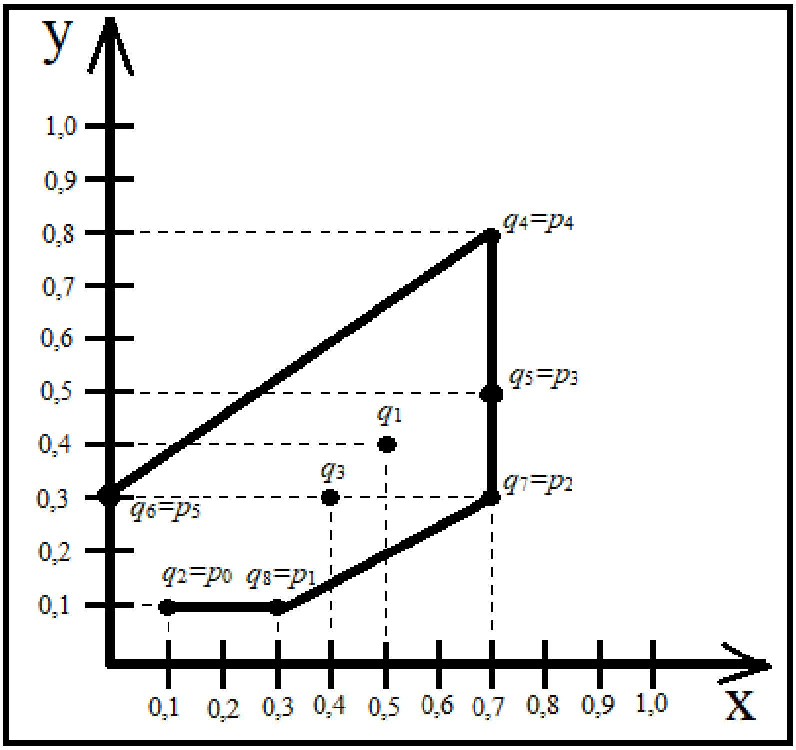
Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_6(0;\ 0{,}3)$.

Покладемо $j = j + 1 = 7$. Так як точка q_7 уже включена до опуклої оболонки, то переходимо до наступної точки.

Покладемо $j = j + 1 = 8$. Так як точка q_8 уже включена до опуклої оболонки, то переходимо до наступної точки.

Усі точки розглянуто, отже, $p_5 = q_6(0; 0{,}3)$.

Включаємо цю точку в оболонку: $Q = \{q_2, q_8, q_7, q_5, q_4, q_6\}$.



Крок 6. Знаходимо наступну за q_6 точку на опуклій оболонці.

Покладемо $\boldsymbol{i} = \mathbf{6}$. В якості претендента візьмемо $\boldsymbol{p_{best}} = \boldsymbol{q_1}\mathbf{(0{,}5;\ 0{,}4)}$, тоді порівняння починаємо з другої точки.

Покладемо $j = 2$. Порівняємо $q_j = q_2(0{,}1; 0{,}1)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_5) \times (q_2 - p_5) = (0{,}5; 0{,}1) \times (0{,}1; -0{,}2) = \det\begin{pmatrix} 0{,}5 & 0{,}1 \\ 0{,}1 & -0{,}2 \end{pmatrix} = -0{,}10 - 0{,}01 = -0{,}11 < 0.$

Так як векторний добуток від'ємний, замінюємо p_{best}: $p_{best} = q_2(0{,}1; 0{,}1)$.

Покладемо $j = j + 1 = 3$. Порівняємо $q_j = q_3(0{,}4; 0{,}3)$ з p_{best}

Запишемо векторний добуток $(p_{best} - p_5) \times (q_3 - p_5) = (0{,}1; -0{,}2) \times (0{,}4; 0) = \det\begin{pmatrix} 0{,}1 & 0{,}4 \\ -0{,}2 & 0 \end{pmatrix} = 0 + 0{,}08 = 0{,}08 > 0.$

Так як векторний добуток додатній, то точка p_{best} зберігається.

Покладемо $j = j + 1 = 4$. Так як точка q_4 уже включена до опуклої оболонки, то переходимо до наступної точки.

Покладемо $j = j + 1 = 5$. Так як точка q_5 уже включена до опуклої оболонки, то переходимо до наступної точки.

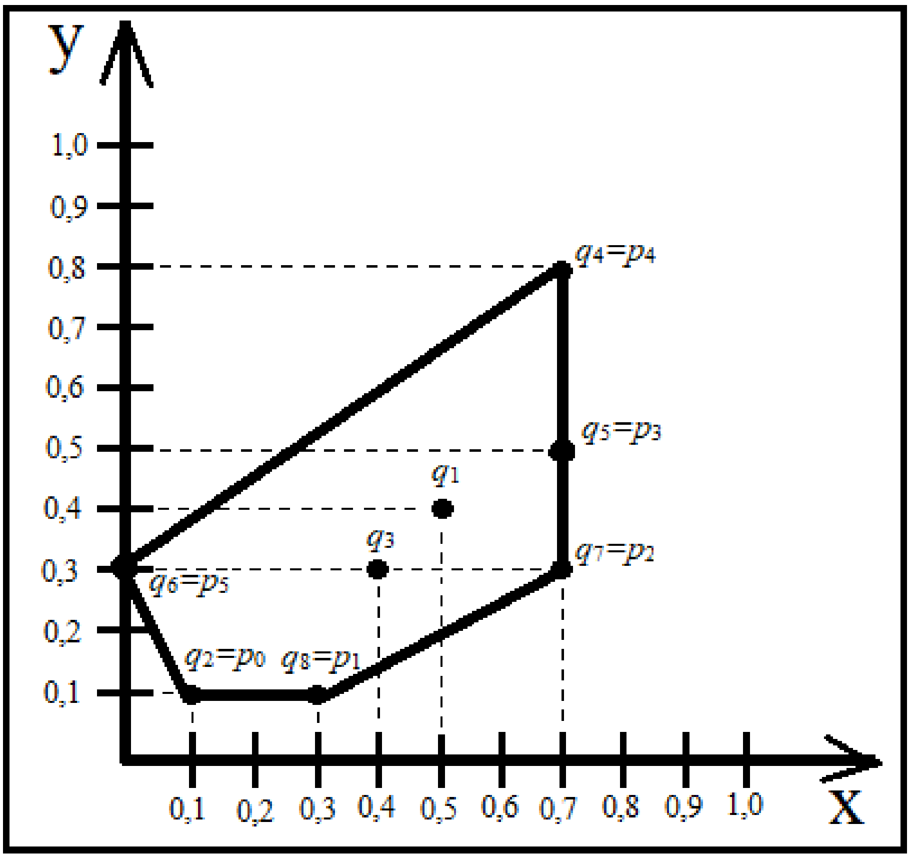
Покладемо $j = j + 1 = 6$. Так як точка q_6 уже включена до опуклої оболонки, то переходимо до наступної точки.

Покладемо $j = j + 1 = 7$. Так як точка q_7 уже включена до опуклої оболонки, то переходимо до наступної точки.

Покладемо $j = j + 1 = 8$. Так як точка q_8 уже включена до опуклої оболонки, то переходимо до наступної точки.

Усі точки розглянуто, а так як отримана на даному етапі точка-претендент співпадає з початковою точкою q_2, то оболонка повністю побудована:

$$Q = \{q_2, q_8, q_7, q_5, q_4, q_6\}.$$



Розв'язання №2

$p_0 = (0{,}1; 0{,}1)$

$p_1 = (0{,}3; 0{,}1)$

$p_2 = (0{,}7; 0{,}3)$

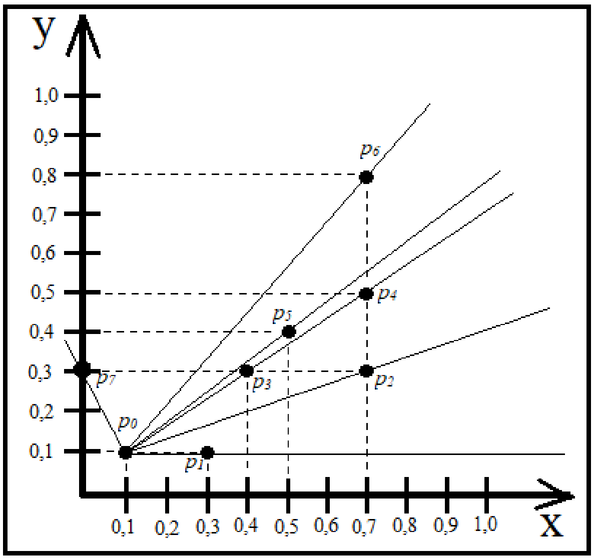
$p_3 = (0{,}4; 0{,}3)$

$p_4 = (0{,}7; 0{,}5)$

$p_5 = (0{,}5; 0{,}4)$

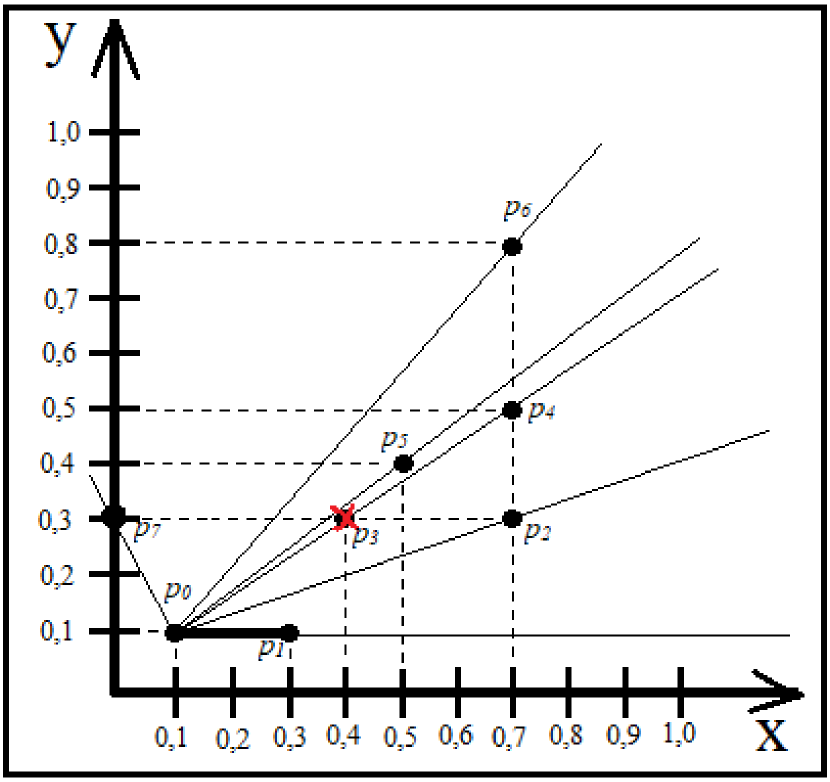
$p_6 = (0{,}7; 0{,}8)$

$p_7 = (0; 0{,}3)$



Для побудови опуклої оболонки будемо використовувати алгоритм обходу Грехема. Відсортуємо точки в порядку зростання полярних кутів відносно стартової точки p_0.

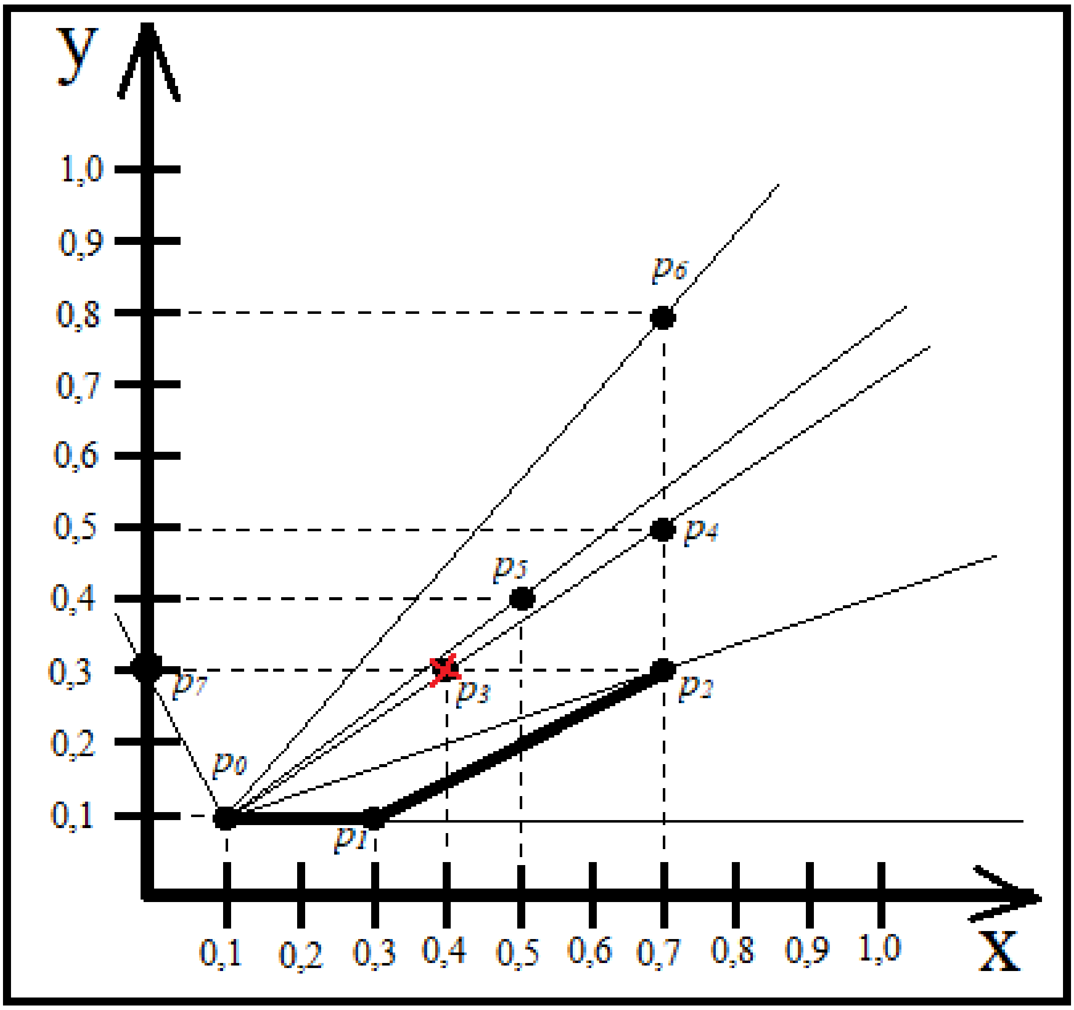
Крок 0. Так як точки p_3 та p_4 мають однаковий полярний кут, то залишаємо для подальшого розгляду точку, відстань від якої до точки p_0 більша. В нашому випадку це точка p_4, тому виключаємо точку p_3 з розгляду. Включаємо до оболонки стартову точку p_0, а також першу точку з відсортованого списку: $Q = \{p_0, p_1\}$.



Крок 1. Розглядаємо дві точки з оболонки, що формується, p_0, p_1 і наступну не розглянуту точку з відсортованого списку – p_2. Визначаємо напрям повороту з точки p_1 в точку p_2, для чого записуємо векторний добуток:

$$(p_1 - p_0) \times (p_2 - p_0) = (0{,}2; 0) \times (0{,}6; 0{,}2) = \det\begin{pmatrix} 0{,}2 & 0{,}6 \\ 0 & 0{,}2 \end{pmatrix} = 0{,}04 + 0 =$$
$$= 0{,}04 > 0.$$

Так як векторний добуток додатній, то поворот лівий, тобто з точки p_1 в точку p_2 необхідно йти наліво відносно прямої, яка визначається точками p_0 і p_1. Значить, включаємо до оболонки точку p_2: $Q = \{p_0, p_1, p_2\}$.

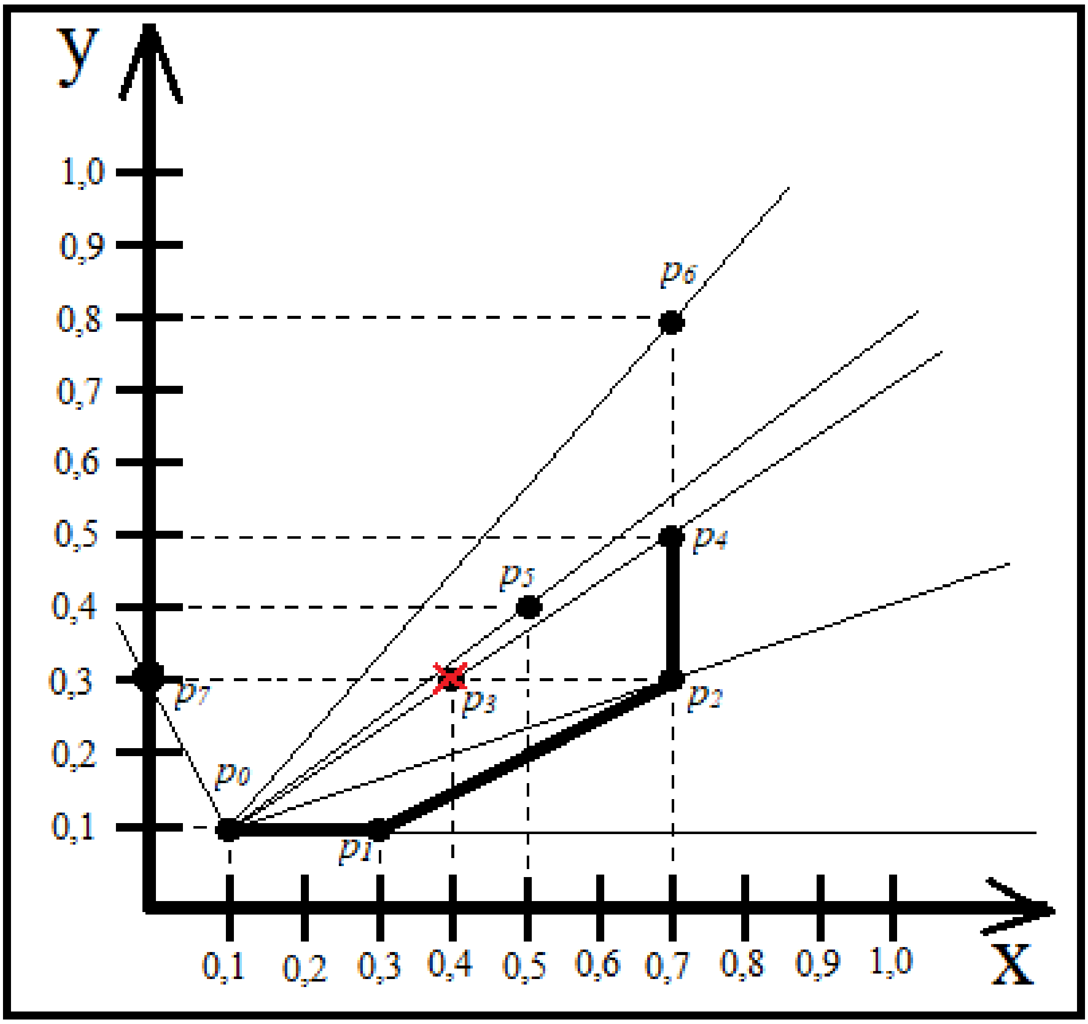


Крок 2. Рухаємося далі по списку. Розглядаємо точки p_1, p_2 з оболонки та точку p_4. Запишемо векторний добуток:

$$(p_2 - p_1) \times (p_4 - p_1) = (0{,}4; 0{,}2) \times (0{,}4; 0{,}4) = \det\begin{pmatrix} 0{,}4 & 0{,}4 \\ 0{,}2 & 0{,}4 \end{pmatrix} =$$

$$= 0{,}16 - 0{,}08 = 0{,}08.$$

Так як векторний добуток додатній, то поворот лівий, тобто з точки p_2 в точку p_4 необхідно йти наліво відносно прямої, яка визначається точками p_1 і p_2. Значить, включаємо до оболонки точку p_4: $Q = \{p_0, p_1, p_2, p_4\}$.

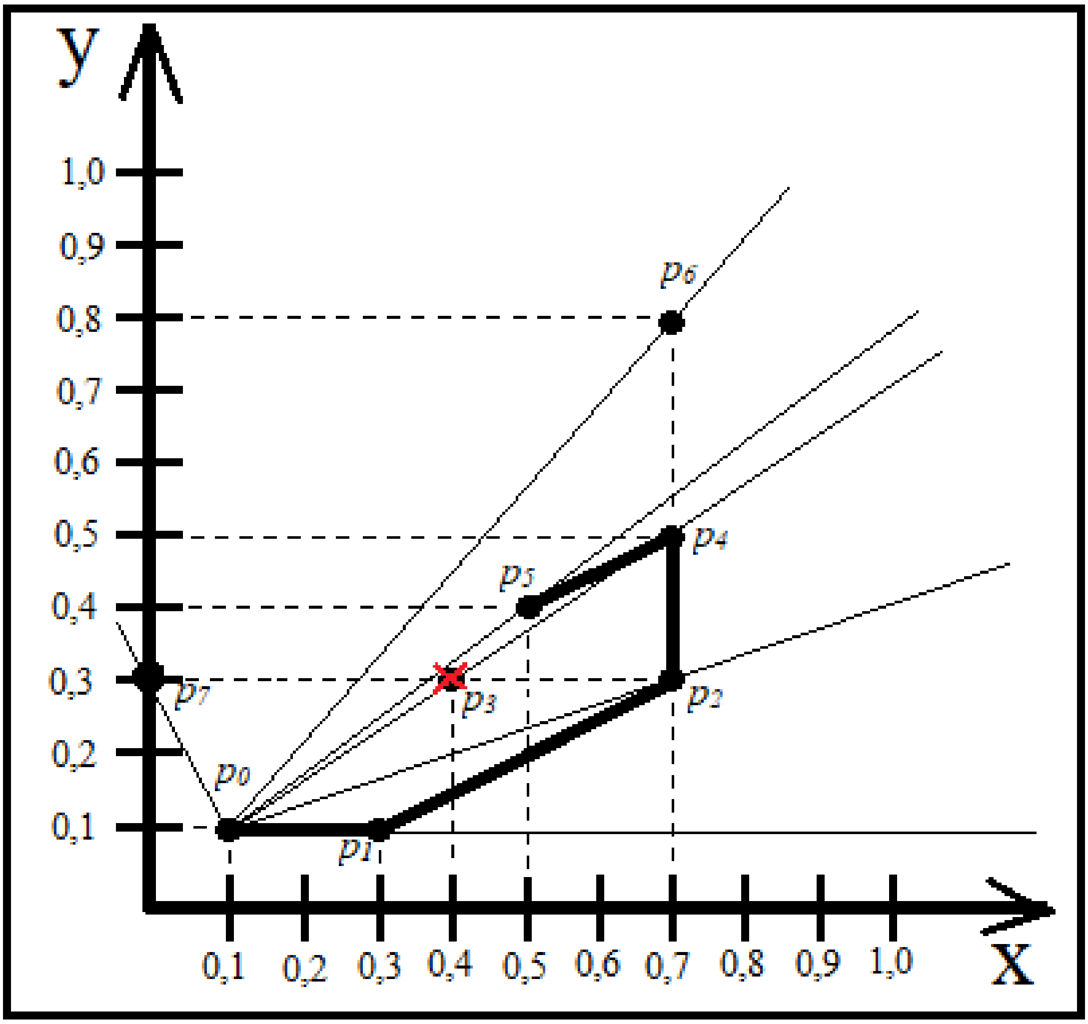


Крок 3. Рухаємося далі по списку. Розглядаємо точки p_2, p_4 з оболонки та не розглянуту точку p_5. Запишемо векторний добуток:

$$(p_4 - p_2) \times (p_5 - p_2) = (0; 0{,}2) \times (-0{,}2; 0{,}1) = \det\begin{pmatrix} 0 & -0{,}2 \\ 0{,}2 & 0{,}1 \end{pmatrix} =$$

$$= 0 + 0{,}04 = 0{,}04.$$

Так як векторний добуток додатній, то поворот лівий, тобто з точки p_4 в точку p_5 необхідно йти наліво відносно прямої, яка визначається точками p_2 і p_4. Значить, включаємо до оболонки точку p_5: $Q = \{p_0, p_1, p_2, p_4, p_5\}$.

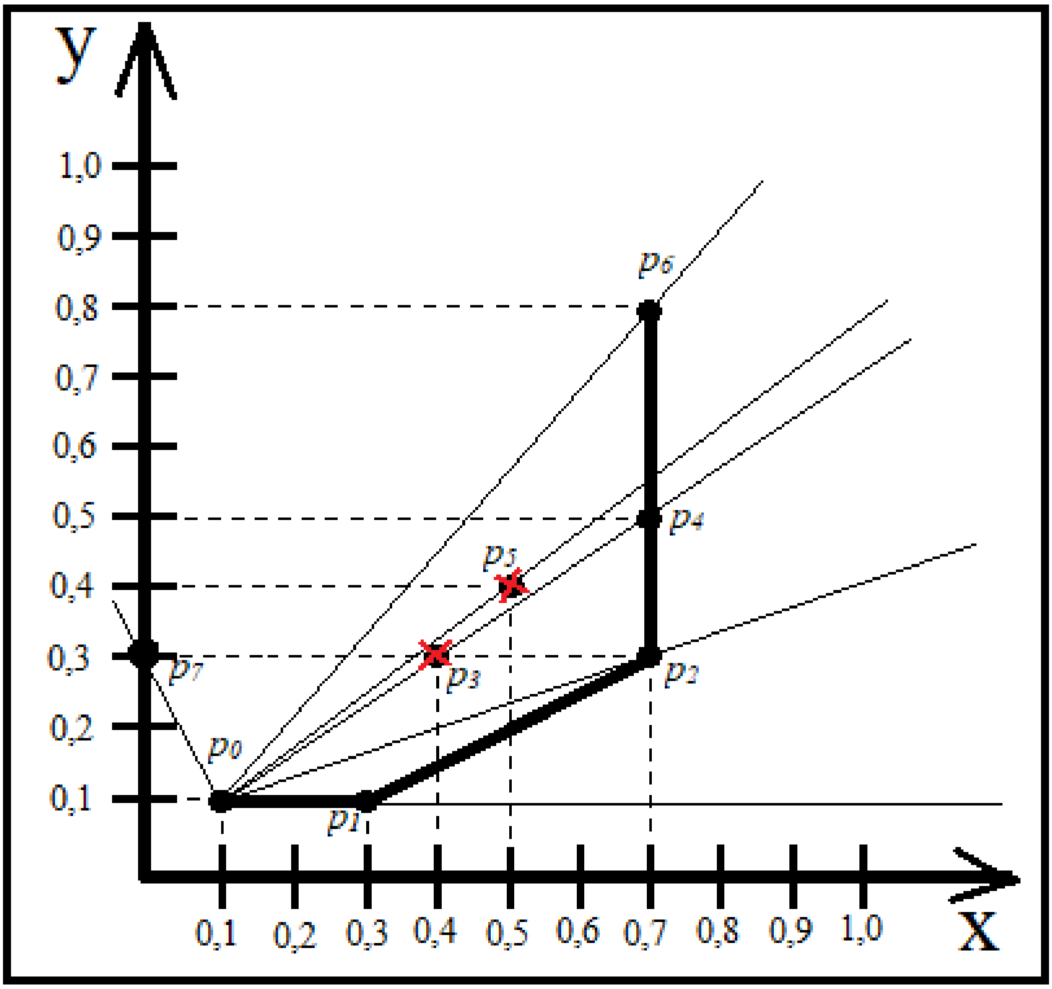


Крок 4. Рухаємося далі по списку. Розглядаємо точки p_4, p_5 з оболонки та не розглянуту точку p_6. Запишемо векторний добуток:

$$(p_5 - p_4) \times (p_6 - p_4) = (-0{,}2; -0{,}1) \times (0; 0{,}3) = \det\begin{pmatrix} -0{,}2 & 0 \\ -0{,}1 & 0{,}3 \end{pmatrix} =$$

$$= -0{,}06 + 0 = -0{,}06.$$

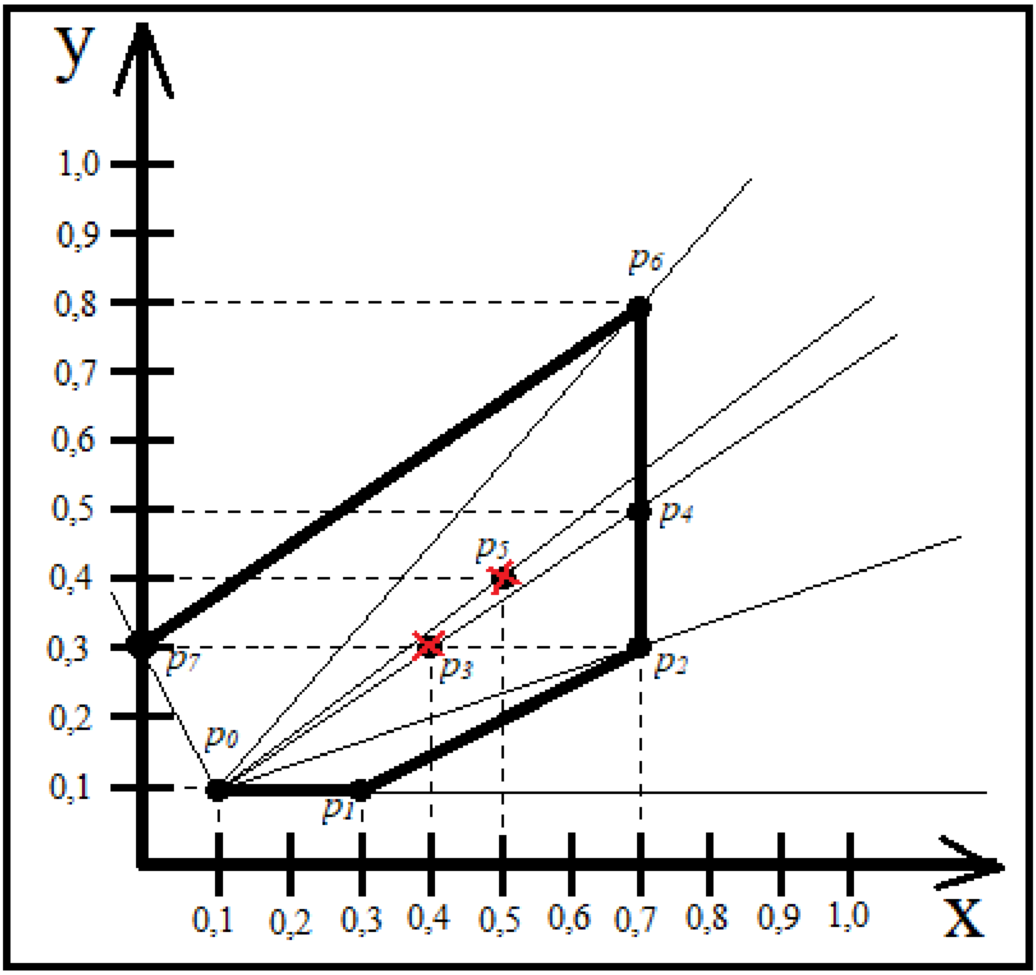
Так як векторний добуток від'ємний, то поворот правий, тобто з точки p_5 в точку p_6 необхідно йти направо відносно прямої, яка визначається точками p_4 і p_5. Значить, виключаємо з оболонки точку p_5 і включаємо до оболонки точку p_6: $Q = \{p_0, p_1, p_2, p_4, p_6\}$.



Крок 5. Рухаємося далі по списку. Розглядаємо точки p_4, p_6 з оболонки та не розглянуту точку p_7. Запишемо векторний добуток:

$$(p_6 - p_4) \times (p_7 - p_4) = (0; 0{,}3) \times (-0{,}7; -0{,}2) = \det\begin{pmatrix} 0 & -0{,}7 \\ 0{,}3 & -0{,}2 \end{pmatrix} =$$
$$= 0 + 0{,}21 = 0{,}21.$$

Так як векторний добуток додатній, то поворот лівий, тобто з точки p_6 в точку p_7 необхідно йти наліво відносно прямої, яка визначається точками p_4 і p_6. Значить, включаємо до оболонки точку p_7: $Q = \{p_0, p_1, p_2, p_4, p_6, p_7\}$.



Усі точки з відсортованого списку розглянуті, значить, оболонку побудовано.