МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДВНЗ «УКРАЇНСЬКИЙ ДЕРЖАВНИЙ ХІМІКО-ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ»

МЕТОДИЧНІ ВКАЗІВКИ
до виконання практичних занять та самостійної роботи з дисципліни “СИСТЕМНИЙ АНАЛІЗ СКЛАДНИХ СИСТЕМ УПРАВЛІННЯ” для студентів IV, V курсів денної, заочної та дистанційної форми навчання спеціальності «151 Автоматизація та комп’ютерно-інтегровані технології»

Затверджено на засіданні кафедри
комп'ютерно-інтегрованих
технологій та метрології.
Протокол №5 від 28.10.2016.

Дніпро ДВНЗ УДХТУ 2016

Методичні вказівки до виконання практичних занять та самостійної роботи з дисципліни “системний аналіз складних систем управління” для студентів IV, V курсів денної, заочної та дистанційної форми навчання спеціальності «151 автоматизація та комп’ютерно-інтегровані технології» / Укл. : Д. А. Лосіхін. – Дніпро : УДХТУ, 2016. – 35 с.

Укладач: Д.А. Лосіхін, старший викладач

Відповідальний за випуск Тараненко Ю.К., професор, докт. техн. наук

Навчальне видання

Методичні вказівки до виконання практичних занять та самостійної роботи з дисципліни “системний аналіз складних систем управління” для студентів IV, V курсів денної, заочної та дистанційної форми навчання спеціальності «151 автоматизація та комп’ютерно-інтегровані технології»

Укладач: ЛОСІХІН Дмитро Анатолійович

Редактор Л.М. Тонкошкур
Коректор Л.Я. Гоцуцова

Підписано до друку 03.10.16. Формат 60х84 1/16. Папір ксерокс. Друк різограф
Умов.-друк. арк. 4,38. Облік.-вид. арк. 4,07. Тираж 30 прим. Замовлення №___
Свідоцтво ДК №303 від 27.12.2000 р.

УДХТУ, 49005, м. Дніпропетровськ, 5, просп. Гагаріна, 8

Видавничо-поліграфічний комплекс ІнКомЦентру

ЗМІСТ

РЕФЕРАТ

Об'єкт. Завдання математичного програмування

Предмет. Рішення завдань в умовах ризику. Рішення завдань в умовах невизначеності. Рішення завдань, заснованих на мережевих моделях. Рішення завдань, заснованих на детермінованих моделях динамічного програмування

Мета. Практична і самостійна робота студента, спрямована на формування знань-копій і знань, що дозволяють вирішувати типові **завдання** по представлених тут темах:

- Ухвалення рішень в умовах ризику (Знати: основні поняття: критерій очікуваного значення, альтернативне рішення, випадкова величина, вірогідність події, стани природи, дерево рішень, вирішальна і випадкова вершини. Уміти: синтезувати дерево рішень завдань в умовах ризику на основі аналізу можливих альтернативних рішень, формувати критерій очікуваного значення, використовувати табличну форму критерію очікуваного значення для вибору оптимального рішення)
- Ухвалення рішень в умовах невизначеності (Знати: основні поняття: матриця платежів, міра консерватизму, принцип недостатньої основи, критерії: Лапласа, мінімаксний, максимінний, Сэвиджа, Гурвиця, показник оптимізму. Уміти: застосовувати критерії Лапласа, мінімаксний, Сэвиджа, Гурвиця на початковій матриці платежів для ухвалення оптимального рішення завдань в умовах невизначеності)
- Мережеві моделі (Знати: основні поняття: мережа, вузол (вершина), дуга (ребро), потік, орієнтоване ребро, шлях, цикл, зв'язна мережа, дерево, остовное дерево. Уміти: застосовувати алгоритми побудови мінімального остовного дерева і визначення найкоротшого шляху (алгоритм Флойда) для оптимального вирішення завдань, заснованих на мережевих моделях)
- Детерміновані моделі динамічного програмування (Знати: принцип динамічного програмування; принцип декомпозиції складного завдання; принцип оптимальності; принцип рекурентних обчислень. Уміти: застосовувати основні елементи моделей динамічного програмування; застосовувати алгоритми прямого і зворотного прогону для вирішення складних завдань)

Контроль засвоєння матеріалу, що вивчається, проводиться на наступному практичному занятті і припускає обговорення виниклих питань і захист контрольних завдань по відповідній темі.

Примітка. Бажана автоматизація розрахунків з застосуванням програмних засобів MATLAB, Mahtcad, Excel, та ін., або з створенням алгоритмів і програм розрахунків.

ТЕМИ 1, 2. УХВАЛЕННЯ РІШЕНЬ В УМОВАХ РИЗИКУ.
Критерій очікуваного значення; дерево рішень; таблична форма

Якщо рішення приймається в умовах ризику, то вартості альтернативних рішень зазвичай описуються імовірнісними розподілами. Рішення, що з цієї причини приймається, грунтується на використанні критерію очікуваного значення, у відповідності з яким альтернативні рішення порівнюються з точки зору максимізації очікуваного прибутку або мінімізації очікуваних витрат. Такий підхід має свої недоліки, які не дозволяють використовувати його в деяких ситуаціях. Для них розроблені модифікації згаданого критерію і підходи до ухвалення рішень в умовах ризику.

Критерій очікуваного значення

Критерій очікуваного значення зводиться або до максимізації очікуваного (середньою) прибутку, або до мінімізації очікуваних витрат. В даному випадку передбачається, що прибуток (витрати), пов'язаний з кожним альтернативним рішенням, є випадковою величиною.

Дерево рішень (1-а модифікація критерію очікуваного значення)

У наведеному нижче простому прикладі розглядається ситуація, пов'язана з ухваленням рішення за наявності кінцевого числа альтернатив і точних значень матриці прибутків.

Приклад 1

Припустимо, необхідно вкласти на фондовій біржі 10000 у.о. в акції однієї з двох компаній: А або В. Акції компанії А являються ризикованими, але можуть принести 50% прибутку від суми інвестиції на протязі наступного року. Якщо умови фондової біржі будуть несприятливі, сума інвестиції може знецінитися на 20%. Компанія В забезпечує безпеку інвестицій з 15% прибутку в умовах підвищення котирувань на біржі і тільки 5% - в умовах пониження котирувань. Усі аналітичні публікації з вірогідністю 60% прогнозують підвищення котирувань і з вірогідністю 40% — зниження котирувань. У яку компанію слід вкласти інвестиції?

Інформація, пов'язана з ухваленням рішення, підсумовувана у наступній таблиці.

	Прибуток від інвестиції за один рік	
Альтернативне рішення	При підвищенні котирувань, (у.о.)	При зниженні котирувань, (у.о.)
Акції компанії А	5000	-2000
Акції компанії В	1500	500
Вірогідність події	0.6	0.4

Це завдання може бути представлене у вигляді дерева рішень, показаного на рис. 1. На цьому рисунку використовуються два типи вершин: квадрат представляє "вирішальну" вершину, а круг - "випадкову". Таким чином, з вершини 1 виходять дві гілки, що представляють альтернативи, пов'язані з купівлею акцій компанії А або В. Далі дві гілки, що виходять з вершин 2 і 3, відповідають випадкам підвищення і зниження котирувань на біржі з вірогідністю їх появи і відповідними платежами.

Виходячи з схеми рис. 1, отримуємо очікуваний прибуток за рік для кожної з двох альтернатив.

Для акцій компанії *A*: 5000·0.6 + (-2000)·0.4=2200 у.о.

Для акцій компанії *B*: 1500·0.6 + 500·0.4 = 1100 у.о.

Вашим рішенням, заснованим на цих обчисленнях, є купівля акцій компанії А.

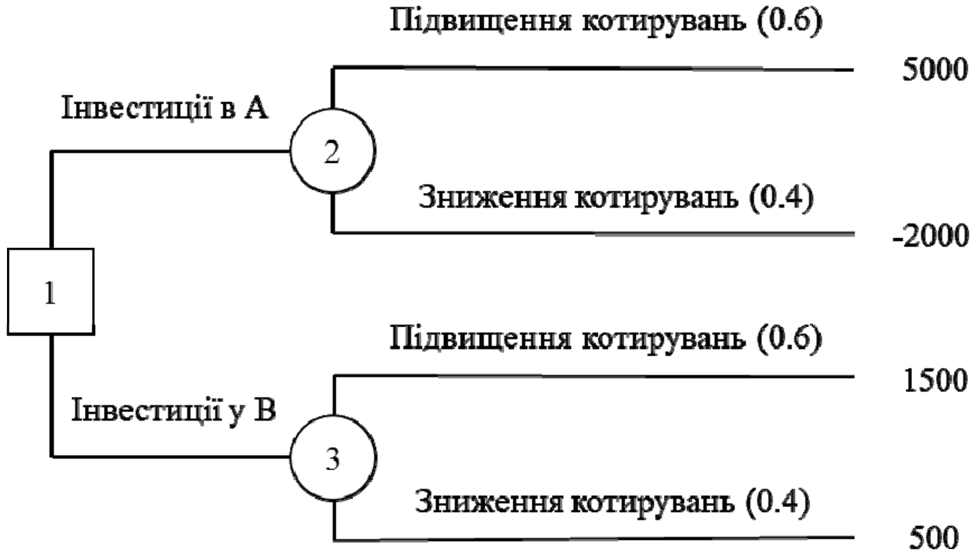


Рис. 1 – Дерево рішень

У теорії ухвалення рішень підвищення і пониження котирувань на біржі іменуються станами природи, можливі реалізації яких є випадковими подіями (в даному випадку з вірогідністю 0.6 і 0.4). У загальному випадку завдання прийняття решений може включати *n* станів природи і *m* альтернатив. Якщо p_j – вірогідність *j*-го стану природи, а a_{ij} — платіж, пов'язаний з ухваленням рішення *i* при стані природи *j* ($i = \overline{1,m}$, $j = \overline{1,n}$), тоді очікуваний платіж для вирішення *i* обчислюють у виді

$$D_i = a_{i1}p_1 + a_{i2}p_2 + ... + a_{in}p_n, \ i = \overline{1,n},$$

де за визначенням $\sum_{i=1}^{n} p_i = 1$.

Найкращим рішенням буде те, яке відповідає $D_i^* = \max_i \{D_i\}$ чи $D_i^* = \min_i \{D_i\}$, залежно від того, чи являється платіж в завданні доходом (прибутком) або збитком (витратами).

Таблична форма (2-а модифікація критерію очікуваного значення)

Розглянемо ситуацію прийняття рішень, в яких плата є математичною функцією альтернативних рішень. В цьому випадку представлення завдання у вигляді дерева рішень хоча і є можливим, але може бути не таким корисним, як в попередньому прикладі.

Приклад 2

Метрологічна служба (МС) підприємства використовує 200 засобів вимірювань (ЗВ). МС планує періодичну перевірку ЗВ. Вірогідність p_t непридатності ЗВ до експлуатації після закінчення t місяців після перевірки оцінюється таким чином.

t	1	2	3	4	5	6	7	8	9	>10
p_t	0.05	0.07	0.10	0.13	0.18	0.23	0.33	0.43	0.50	0.55

Випадкова метрологічна відмова одного ЗВ обходиться підприємству в 100 у.о., а планована періодична перевірка в 10 у.о. Необхідно визначити оптимальний період (у місяцях) між планованими періодичними перевірками.

Позначимо через N шукане число місяців між періодичними перевірками. Упродовж N-месячного циклу можуть мати місце два види витрат: 1) витрати, пов'язані з усуненням метрологічної відмови ЗВ упродовж перших N-1 місяців і 2) витрати на періодичну перевірку у кінці циклу. Витрати другого виду (періодична перевірка) складають 10х200 ЗВ, тобто 2000 у.о. на цикл. Витрати, пов'язані з усуненням метрологічної відмови ЗВ, повинні грунтуватися на середній кількості ЗВ, що вийшли з ладу упродовж перших N-1 місяців циклу. Тут ми маємо два стани після закінчення місяця t: метрологічна відмова з вірогідністю p_t і його відсутність з вірогідністю $1-p_t$. Отже, очікуване число метрологічних відмов після закінчення місяця t дорівнює кількості ЗВ на підприємстві, помноженому на p_t, тобто $200p_t$. Використовуючи цей результат, підрахуємо очікуване загальне число ЗВ, що відмовили, упродовж перших N-1 місяців циклу у вигляді суми відповідних величин для кожного місяця окремо, тобто $200p_1 + 200p_2 + ... + 200p_{N-1} = 200(p_1+p_2 +...+p_{N-1})$. Позначивши через CN загальну очікувану вартість для циклу між періодичними перевірками, маємо наступне.

CN = 2000 + 100*200($p_1+p_2+...+p_{N-1}$), у.о.

Завдання вибору рішення МС зводиться таким чином до визначення довжини циклу N, яка мінімізує загальні очікувані витрати за один місяць CMN, тобто величину $CMN = \frac{CN}{N} = \frac{2000 + 20000\sum_{i=1}^{N-1} p_i}{N}$.

Мінімізацію функції CMN не можна виконати в явній формі. Замість цього використовується наступна таблична форма знаходження рішення.

	N	p_N	$\sum_{i=1}^{N-1} p_i$	*CMN*
	1	0.05	0.00	2000.00
	2	0.07	0.05	1500.00
Оптимальне *N*	3	0.10	0.12	1466.67
	4	0.13	0.22	1600.00
	5	0.18	0.35	1800.00
	6	0.23	0.53	2100.00
	7	0.33	0.76	2457.14

Обчислення показують, що CMN досягає свого мінімуму при N=3. Отже, періодичну перевірку ЗВ треба виконувати кожні три місяці.

Завдання вибору рішення в прикладі 2 можна також представити у вигляді дерева рішень.

ЗАВДАННЯ НА САМОСТІЙНУ РОБОТУ

Завдання 1. У завданні з прикладу 2 вартість періодичної перевірки одного ЗВ дорівнює x_1=20 у.о., а вартість усунення метрологічної відмови — x_2=40 у.о. Вірогідність метрологічної відмови ЗВ в перший місяць дорівнює p_1=0.03 і збільшується на Δp=0.01 для кожного наступного місяця, по десятий включно. Починаючи з одинадцятого місяця і далі, вірогідність метрологічної відмови зберігається постійною на рівні p_2=0.13.

а) Побудуйте відповідне дерево рішень.
б) Визначите оптимальну довжину циклу для періодичної перевірки.

Завдання 2. Щорічний попит на ЗВ певного виду в мережі продажів задається слідуючим розподілом вірогідності.

n	100	150	200	250	300
P_n	0.20	0.25	0.30	0.15	0.10

Фірма купує ЗВ по a=500 у.о., а продає по b=1000 у.о. Якщо ЗВ не проданий в той же рік, то до кінця року ЗВ може бути реалізованим за c=250 у.о. Величина запасу ЗВ може приймати одне з можливих значень попиту, які перелічені вище.

а) Побудуйте відповідне дерево рішень.
б) Скільки ЗВ необхідно закуповувати щорічно?

Завдання 3. Нехай в попередньому завданні часовий інтервал, для якого необхідно вирішити завдання ухвалення рішень, складає два роки.

Альтернативи для другого року залежать від об'єму реалізації ЗВ в перший рік. Якщо реалізований у точності увесь запас першого року, фірма замовить таку ж кількість ЗВ і на другий рік. Якщо потреба в ЗВ в перший рік перевищує наявний запас, то для другого року фірма може замовити будь-який з об'ємів попиту на ЗВ, який перевищує запас першого року. І нарешті, якщо в перший рік реалізовано менше ЗВ, чим було закуплено, то для другого року фірма може замовити будь-який з об'ємів попиту на ЗВ, який менше запасу першого року. Побудуйте відповідне дерево рішень і визначте оптимальну стратегію замовлення.

а) Побудуйте відповідне дерево рішень.
б) Визначите оптимальну стратегію замовлення.

Завдання 4. Автомат призводить *a* тисяч одиниць деякого продукту щодня. Якщо *a* збільшується, доля браку *p,* будучи випадковою величиною, зростає відповідно до наступної функції щільності розподілу:

$$f(p) = \begin{cases} a \cdot p^{a-1}, \text{при } 0 \le p \le 1 \\ 0, \text{при } 0 > p > 1 \end{cases}$$

Кожен бракований виріб приносить збиток в x=40 у.о., а якісний виріб - прибуток y=5 у.о.

а) Поясніть, чому важко побудувати дерево рішень для цього завдання.
б) Визначте значення *a*, при якому очікуваний прибуток набуває максимального значення.

ВАРІАНТИ ПОЧАТКОВИХ ДАНИХ (теми 1, 2)

початкові дані	Завдання 1					Завдання 2			Завдання 4	
	параметр завдання									
№ варіанту	x_1	x_2	p_1	p_2	Δp	a	b	c	x	y
0	14	36	0.02	0.11	0.01	240	300	160	15	5
1	8	16	0.02	0.10	0.01	200	300	70	47	7
2	9	27	0.03	0.12	0.01	260	400	100	95	20
3	10	32	0.02	0.11	0.01	300	500	180	33	3
4	11	19	0.03	0.13	0.01	256	1024	128	60	16
5	12	38	0.02	0.14	0.01	90	200	30	50	10
6	13	42	0.04	0.15	0.01	512	1024	256	77	12
7	14	49	0.02	0.11	0.01	240	500	175	80	15
8	15	38	0.03	0.10	0.01	600	990	370	50	4
9	16	50	0.02	0.12	0.01	450	800	330	18	3
10	17	54	0.01	0.13	0.01	700	1000	400	45	6
11	18	80	0.02	0.15	0.01	400	680	250	70	17
12	19	73	0.03	0.16	0.01	370	510	150	60	8
13	20	68	0.02	0.12	0.01	430	780	300	89	14
14	21	83	0.04	0.11	0.01	390	700	150	49	8
15	12	66	0.01	0.10	0.01	150	270	50	34	6
16	8	20	0.02	0.10	0.01	210	300	80	17	5
17	9	32	0.01	0.12	0.02	280	400	140	26	8
18	10	30	0.02	0.11	0.01	320	500	180	37	14
19	11	28	0.01	0.13	0.02	260	320	200	56	7
20	12	40	0.02	0.14	0.01	90	160	40	17	9
21	13	21	0.01	0.15	0.02	512	600	280	29	9
22	15	41	0.01	0.10	0.02	600	800	380	37	13
23	16	37	0.03	0.12	0.01	450	700	340	17	7
24	17	42	0.01	0.13	0.02	700	800	300	19	11
25	18	47	0.02	0.15	0.01	400	580	180	53	35
26	19	50	0.01	0.16	0.02	370	420	165	24	13
27	20	48	0.04	0.12	0.01	410	530	300	35	17
30	21	71	0.01	0.10	0.01	160	210	50	76	37
31	22	65	0.02	0.10	0.01	410	490	185	54	27
32	23	59	0.01	0.12	0.02	570	700	330	35	15
33	24	74	0.02	0.11	0.01	610	750	300	27	12
34	8	16	0.01	0.12	0.02	500	725	200	28	9
35	9	32	0.02	0.13	0.01	215	310	100	33	19
36	10	17	0.01	0.12	0.02	80	115	50	14	3
37	11	21	0.02	0.14	0.01	210	299	100	17	8
38	12	19	0.01	0.11	0.02	490	560	300	21	7
39	13	33	0.02	0.10	0.01	700	750	600	30	13
30	21	71	0.01	0.10	0.01	160	210	50	34	17

ТЕМИ 3, 4. УХВАЛЕННЯ РІШЕНЬ В УМОВАХ НЕВИЗНАЧЕНОСТІ. Критерій Лапласа; критерій максимінний; критерій Севиджа; критерій Гурвиця

Ухвалення рішень в умовах невизначеності вимагає визначення альтернативних дій, яким відповідають платежі, залежні від (випадкових) *станів природи*.

Матрицю платежів в завданні ухвалення рішень з m можливими діями і n станами природи можна представити таким чином.

$$
\begin{array}{c|cccccc}
 & s_1 & s_2 & \dots & s_j & \dots & s_n \\
\hline
a_1 & v(a_1,s_1) & v(a_1,s_2) & \dots & v(a_1,s_j) & \dots & v(a_1,s_n) \\
a_2 & v(a_2,s_1) & v(a_2,s_2) & \dots & v(a_2,s_j) & \dots & v(a_2,s_n) \\
\dots & \dots & \dots & \dots & & \dots & \dots \\
a_i & v(a_i,s_1) & v(a_i,s_2) & \dots & v(a_i,s_j) & \dots & v(a_i,s_n) \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
a_m & v(a_m,s_1) & v(a_m,s_2) & \dots & v(a_m,s_j) & \dots & v(a_m,s_n)
\end{array}
$$

Елемент a_i представляє i-е можливе рішення, а елемент s_j — j-е стан природи. Плата (чи дохід), пов'язана з рішенням a_i і станом s_j дорівнює $v(a_i,s_j)$.

В умовах невизначеності імовірнісний розподіл, що відповідає станам s_j або невідомий, або не може бути визначеним. В цьому випадку аналіз ухвалення рішення можливий із застосуванням наступних критеріїв.

Критерій Лапласа спирається на принцип недостатньої основи, який свідчить, що оскільки розподіл вірогідності станів $P(s_j)$ невідомий, немає причин вважати їх різними. Отже, використовується оптимістичне припущення, що вірогідність усіх станів природи рівна між собою, тобто $P\{s_1\}= P\{s_1\}=\dots P\{s_n\}=1/\text{n}$. Якщо при цьому $v(a_i,s_j)$ представляє отримуваний прибуток, то найкращим рішенням є те, яке забезпечує

$$\max_{a_i}\left\{\frac{1}{n}\sum_{j=1}^{n}v(a_i,s_j)\right\}$$

Якщо величина $v(a_i,s_j)$ представляє витрати *особи, що приймає рішення* (*ОПР*), то оператор "max" замінюється на "min".

Максимінний (мінімаксний) критерій заснований на консервативній обережній поведінці особи, що приймає рішення, і зводиться до вибору найкращої альтернативи з найгірших. Якщо величина $v(a_i,s_j)$ представляє отримуваний прибуток (витрати), то відповідно до максимінного (мінімаксного) критерію в якості оптимального вибирається відповідно рішення, що забезпечує

$$\max_{a_i}\left\{\min_{s_j} \nu(a_i,s_j)\right\} \text{ чи } \min_{a_i}\left\{\max_{s_j} \nu(a_i,s_j)\right\}.$$

Критерій Сэвиджа прагне пом'якшити консерватизм мінімаксного (максимінного) критерію шляхом заміни матриці платежів (виграшів або програшів) $\nu(a_i,s_j)$ матрицею втрат $r(a_i,s_j)$, яка визначається таким чином:

$$r(a_i,s_j)=\begin{cases}\left\{\max\limits_{a_k} \nu(a_k,s_j)\right\}-\nu(a_i,s_j), \nu \text{ - доход}\\ \nu(a_i,s_j)-\left\{\min\limits_{a_k} \nu(a_k,s_j)\right\}, \nu \text{ - втрати}\end{cases}$$

Критерій Гурвіця охоплює ряд різних підходів до ухвалення рішень - від найбільш оптимістичного до найбільш песимістичного. Нехай $0 \le \alpha \le 1$ і величини $\nu(a_i,s_j)$ представляють прибутки (збитки). Тоді рішенню, вибраному за критеріэм Гурвіця відповідає

$$\max_{a_i}\left\{\alpha \max_{s_j} \nu(a_i,s_j)+(1-\alpha)\min_{s_j} \nu(a_i,s_j)\right\}$$

чи

$$\min_{a_i}\left\{\alpha \min_{s_j} \nu(a_i,s_j)+(1-\alpha)\max_{s_j} \nu(a_i,s_j)\right\}$$

Параметр α - показник оптимізму. Якщо α=0, критерій Гурвіця стає консервативним, оскільки його застосування еквівалентно застосуванню звичайного мінімаксного критерію. Якщо α=1, критерій Гурвіця стає занадто оптимістичним, бо розраховує на найкращі з найкращих умов. Ми можемо конкретизувати міру оптимізму (чи песимізму) належним вибором величини α з інтервалу [0,1]. За відсутності яскраво вираженої схильності до оптимізму або песимізму вибір α=0.5 представляється найбільш розумним.

Приклад 1

Служба (КВПтаА) підприємства вирішує організувати необхідну кількість АРМ наладчиків РЕА. Число ТЗА, що підлягають ремонту, у місяць відповідно до графіку поточного ремонту може бути 40, 60, 80, 100.

Відхилення у бік зменшення або збільшення кількості АРМ спричиняють за собою додаткові витрати, обумовлені будівництвом надмірних (невживаних) АРМ або втратою можливості отримати прибуток у разі, коли потреби підприємства в справних ТЗА не задовольняються. Нехай змінні a_1-a_4 представляють можливі кількості АРМ (з розрахунку на обслуговування 40, 60, 80 і 100 ТЗА в місяць), а змінні s_1-s_4 - відповідне число ТЗА, що поступають в ремонт. Наступна таблиця містить матрицю вартостей (у.о.) в цій ситуації.

	s_1	s_2	s_3	s_4
a_1	6	10	18	25
a_2	9	7	14	21
a_3	20	16	9	24
a_4	30	25	20	12

Проведемо аналіз цієї ситуації, використовуючи описані вище критерії.

Критерій Лапласа

При заданих вірогідностях $P\{s_j)=1/n=0.25$ очікувані значення витрат для різних можливих рішень обчислюються таким чином:

$$M\{a_1\}=0.25*(6+10+18+25) \quad =14.75 \text{ у.о.}$$
$$M\{a_2\}=0.25*(9+7+14+21) \quad =12.75 \text{ у.о.} \quad \leftarrow \text{оптимум}$$
$$M\{a_3\}=0.25*(20+16+9+24) \quad =17.25 \text{ у.о.}$$
$$M\{a_4\}=0.25*(30+25+20+12) \quad =21.75 \text{ у.о.}$$

Критерій мінімаксний

Цей критерій використовує початкову матрицю вартостей.

	s_1	s_2	s_3	s_4	Максимум рядків	
a_1	6	10	18	25	25	
a_2	9	7	14	21	21	← мінімакс
A_3	20	16	9	24	24	
A_4	30	25	20	12	30	

Критерій Сэвиджа

Матриця втрат в даному випадку визначається за допомогою віднімання елементів діагоналі початкової матриці вартостей з елементів стовпців від першого до четвертого відповідно. Тоді

	s_1	s_2	s_3	s_4	Максимум рядків	
A_1	0	3	9	13	13	
A_2	3	0	5	9	9	← мінімакс
a_3	14	9	0	12	14	
a_4	24	18	11	0	24	

Критерій Гурвіця

Результати обчислень містяться в наступній таблиці:

Альтернатива	Мінімум рядків	Максимум рядків	α(мінімум рядка) + (1 -α)(максимум рядка)
a_1	6	25	15.5
a_2	7	21	14.0
a_3	9	24	16.5
a_4	12	30	21.0

Використовуючи бажане значення α, можна визначити оптимальну альтернативу. При α=0.5 оптимальною є альтернатива a_2.

ЗАВДАННЯ НА САМОСТІЙНУ РОБОТУ

Завдання 1. Вирішіть завдання з прикладу 1 (використовуючи кожний з приведених критеріїв), припускаючи, що дохід, пов'язаний з рішенням a_i, і станом s_j, представлений наступною матрицею

	s_1	s_2	s_3	s_4
a_1	6	5	6	12
a_2	7	8	20	20
a_3	9	10	14	17
a_4	12	7	3	30

Примітка. Відповідно до номера варіанту *(x*) і останньої цифри *(y)* шифру студента, кожне значення матриці прибутків коригується вичисленням за формулою $v(a_i, s_j) \cdot x + (x \cdot y)$.

Завдання 2. Один з безлічі N верстатів має бути вибраний для виготовлення Q одиниць продукції. Мінімальна і максимальна потреба в продукції дорівнює $Q_{\min}$ і $Q_{\max}$ відповідно. Виробничі витрати C_i на виготовлення Q одиниць продукції на i-му верстаті включає фіксовані витрати K_i і питомі витрати c_i на виробництво одиниці продукції, і виражаються формулою $C_i = K_i + c_i Q$. Вирішіть завдання за допомогою кожного з чотирьох критеріїв ухвалення рішення в умовах невизначеності, припускаючи, що $Q_{\min} \le Q \le Q_{\max}$. Початкові дані приведені в наступній таблиці

ВАРІАНТИ ПОЧАТКОВИХ ДАНИХ (теми 3, 4)

	початкові дані до завдання 2					
№ варіанту	верстат i	K_i, у.о.	c_i, у.о.	Q_{min}	Q_{max}	α
1	1	100	5	1000	3000	0.1
	2	50	10			
	3	150	3			
	4	80	7			
2	1	120	3	800	2000	0.2
	2	80	8			
	3	160	7			
	4	40	10			
3	1	150	17	300	5000	0.3
	2	90	20			
	3	170	21			
	4	50	14			
4	1	300	23	1200	4700	0.4
	2	200	15			
	3	50	5			
	4	190	9			
5	1	60	9	450	2200	0.6
	2	80	7			
	3	120	8			
	4	75	6			
6	1	110	36	256	2048	0.7
	2	33	48			
	3	177	14			
	4	99	21			
7	1	41	3	1024	4112	0.8
	2	75	5			
	3	114	8			
	4	57	2			

ТЕМИ 5, 6. МЕРЕЖЕВІ МОДЕЛІ.
Алгоритм побудови мінімального остовного дерева; знаходження найкоротшого шляху

Алгоритм побудови мінімального остовного дерева припускає з'єднання усіх вузлів мережі за допомогою шляхів найменшої довжини. Типовим завданням, для вирішення якого потрібний такий алгоритм, являється створення (проектування) мережі доріг з твердим покриттям, що сполучають населені пункти в сільській місцевості, де дороги, що сполучають два які-небудь пункти, можуть проходити через інші населені пункти. Найбільш економічний проект дорожньої системи повинен мінімізувати загальну довжину доріг з твердим покриттям, при цьому бажаний результат можна отримати за допомогою алгоритму побудови мінімального остовного дерева.

Опишемо процедуру виконання цього алгоритму.

Позначимо через N — {1, 2,..., n) безліч вузлів мережі і введемо нові позначення: C_k — безліч вузлів мережі, сполучених алгоритмом після виконання k-ї ітерації цього алгоритму; $\overline{C}_k$ — безліч вузлів мережі, не сполучених з вузлами множини C_k після виконання k-ої ітерації цього алгоритму.

Етап 0. Нехай $C_0 = \varnothing$ і $\overline{C}_0 = N$.

Етап 1. Вибираємо будь-який вузол i з множини C_0 і визначаємо $C_1 = \{i\}$, тоді $\overline{C}_1 = N - \{i\}$. Вважаємо $k = 2$.

Основний етап k. У множині $\overline{C}_{k-1}$ вибираємо вузол j, який сполучений найкоротшою дугою з яким-небудь вузлом з множини C_{k-1}. Вузол j приєднується до множини C_{k-1} і видаляється з множини $\overline{C}_{k-1}$. Таким чином, $C_k = C_{k-1} + \{j\}$, $\overline{C}_k = \overline{C}_{k-1} - \{j\}$.

Якщо множина $\overline{C}_k$ порожня, то виконання алгоритму закінчується. Інакше вважаємо k=k+1 і повторюємо останній етап.

Приклад 1

Телевізійна компанія планує підключення до своєї кабельної мережі п'яти нових районів. На рис. 1 показана структура планованої мережі і відстані (у км) між районами і телецентром. Необхідно спланувати найбільш економічну кабельну мережу.

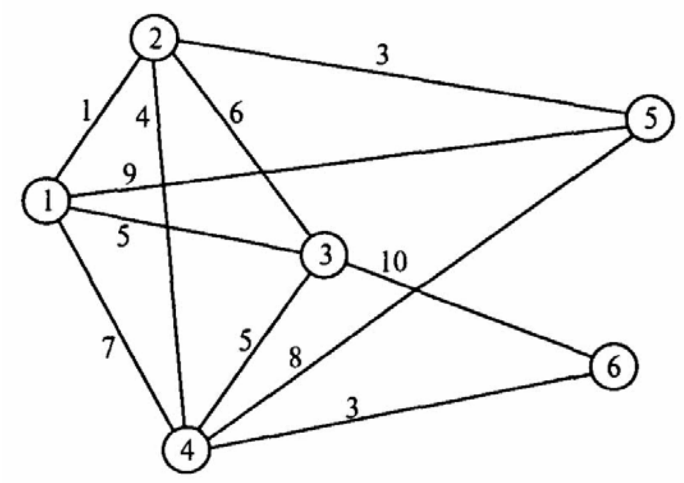
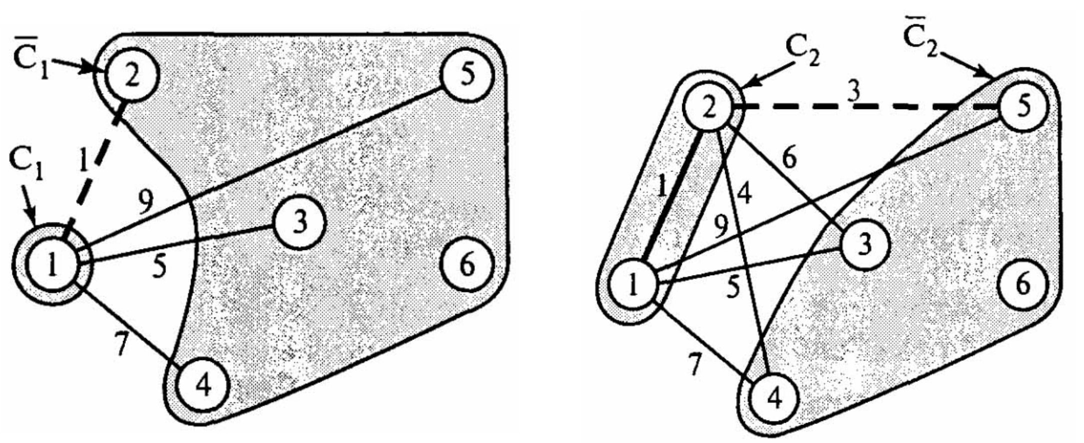


Рис. 1 – Структура планованої мережі

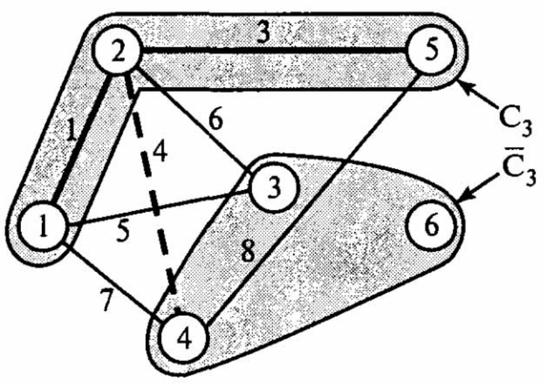
Щоб почати виконання алгоритму побудови мінімального остовного дерева, виберемо вузол 1 (чи будь-який інший вузол). Тоді $C_1 = \{1\}$ і $\overline{C}_1 = \{2,3,4,5,6\}$.

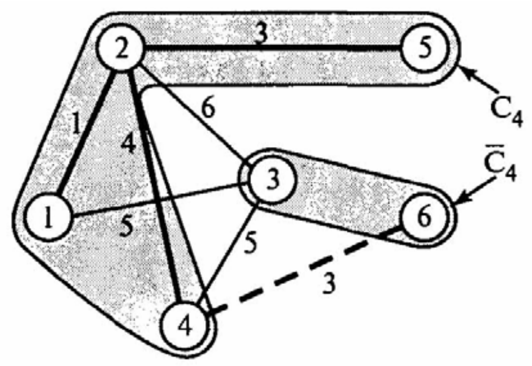
Послідовні ітерації виконання алгоритму представлені на рис. 2. Тут тонкими лініями показані ребра, що сполучають вузли, які належать множинам C_k і $\overline{C}_k$, серед яких шукаємо ребро з мінімальною вартістю (завдовжки). Це знайдене ребро показане пунктирною лінією. Товстими суцільними лініями позначені ребра, що сполучають вузли множини C_k (і які раніше позначалися пунктирними лініями).



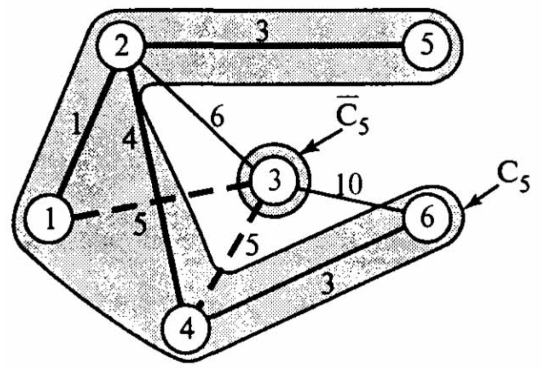
Ітерація 1 Ітерація 2

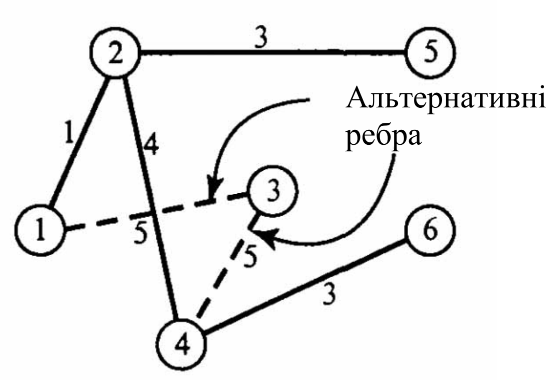
Рис. 2 – Послідовні ітерації виконання алгоритму





Ітерація 3 Ітерація 4





Ітерація 5 Ітерація 6

Рис. 2 – Продовження (послідовні ітерації виконання алгоритму)

Наприклад, на першій ітерації ребро (1, 2) має найменшу вартість (тобто найменша відстань між пунктами мережі) серед усіх інших ребер, що сполучають вузол 1 з вузлами множини $\overline{C}_1$, (відмітимо, що вузол 6 не має ребра, що безпосередньо сполучає його з вузлом 1). Тому j =2 і $C_2 = \{1,2\}$, $\overline{C}_2 = \{3,4,5,6\}$.

Рішення у вигляді мінімального остовного дерева отримане на 6-ій ітерації (рис. 2). Мінімальна довжина кабелю для побудови такої мережі рівна 1+3+4+3+5 = 16 км.

ЗАВДАННЯ НА САМОСТІЙНУ РОБОТУ

Завдання 1. Вирішіть завдання з прикладу 1, починаючи з вузла 5.

Завдання 2. Знайдіть мінімальне остовное дерево для мережі з прикладу 1 при виконанні кожного з наступних умов окремо.

а) Вузли 5 і 6 пов'язані 2-км кабелем.
б) Вузли 2 і 5 не пов'язані.
в) Вузли 2 і 6 пов'язані 4-км кабелем.
г) Вузли 1 і 2 пов'язані кабелем завдовжки 8 км.

д) Вузли 3 і 5 пов'язані кабелем завдовжки 2 км.
е) Вузол 2 не пов'язаний безпосередньо з вузлами 3 і 5.

Завдання знаходження найкоротшого шляху

Це завдання полягає у визначенні в транспортній мережі найкоротшого шляху між заданими початковим пунктом і пунктом призначення.

Алгоритм Флойда дозволяє знаходити найкоротші шляхи між будь-якими двома вузлами мережі. У цьому алгоритмі мережа представлена у вигляді квадратної матриці з n рядками і n стовпцями. Елемент (i, j) дорівнює відстані d_{ij} від вузла i до вузла j, яке має кінцеве значення, якщо існує дуга (i, j), і дорівнює нескінченності інакше.

Нехай є три вузли i, j та k та задані відстані між ними (рис. 3). Якщо виконується нерівність $d_{ij}+d_{jk}<d_{ik}$, те доцільно замінити шлях $i \to k$ шляхом $i \to j \to k$. Така заміна (трикутний оператор) виконується систематично в процесі виконання алгоритму.

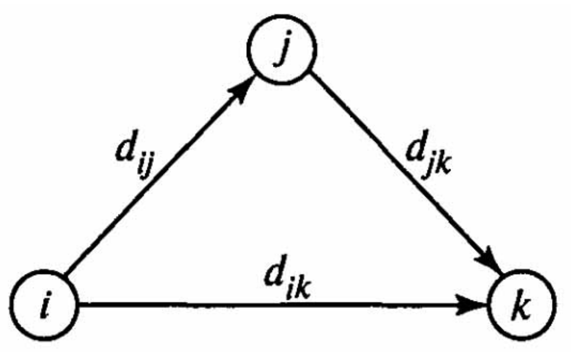


Рис. 3

Алгоритм Флойда вимагає виконання наступних дій:

Етап 0. Визначаємо початкову матрицю відстаней D_0 і матрицю послідовності вузлів S_0. Діагональні елементи обох матриць позначаються знаком "-", який показує, що ці елементи в обчисленнях не беруть участь. Вважаємо k=1.

D_0	1	2	...	j	...	n
1	-	d_{12}	...	d_{1j}	...	d_{1n}
2	d_{21}	-	...	d_{2j}	...	d_{2n}
...	...	...	...	...	...	...
i	d_{i1}	d_{i2}	...	d_{ij}	...	d_{in}
...	...	...	...	...	...	...
n	d_{n1}	d_{n2}	...	d_{nj}	...	-

S_0	1	2	...	j	...	n
1	-	2	...	j	...	n
2	1	-	...	j	...	n
...	...	...	...	...	...	...
i	1	2	...	j	...	n
...	...	...	...	...	...	...
n	1	2	...	j	...	-

Основний етап k. Задаємо рядок k і стовпець k як провідний рядок і провідний стовпець. Розглядаємо можливість застосування трикутного оператора до усіх елементів d_{ij} матриці D_{k-1} Якщо виконується нерівність

$$d_{ik} + d_{kj} < d_{ij}, (i \neq k, j \neq k, i \neq j),$$

те робимо наступне:

а) створюємо матрицю D_k шляхом заміни в матриці D_{k-1} елементу d_{ij} сумою $d_{ik} + d_{kj}$,

б) створюємо матрицю S_k, міняючи в матриці S_{k-1} елемент s_{ij} на k. Вважаємо k=k+1 і повторюємо етап k.

Пояснимо дії, що виконуються на k-му етапі алгоритму, представивши матрицю D_{k-1} так, як вона показана на рис. 4. На цьому малюнку рядок k і стовпець k є ведучими. Рядок i — будь-який рядок з номером від 1 до k-1, а рядок p — довільний рядок з номером від k+1 до n. Аналогічно стовпець j представляє будь-який стовпець з номером від 1 до k-1, а стовпець q — довільний стовпець з номером від k+1 до n. Трикутний оператор виконується таким чином. Якщо сума елементів ведучих рядка і стовпця (показаних в квадратиках) менше елементів, що знаходяться на перетині стовпця і рядка (показані в кухлях), що відповідають даним ведучім елементам, то відстань (елемент в кухлі) замінюється сумою відстаней, представлених ведучими елементами.

Після реалізації n етапів алгоритму визначення по матрицях D_n і S_n найкоротшого шляху між вузлами i та j виконується за наступними правилами:

1. Відстань між вузлами i та j дорівнює елементу d_{ij} у матриці D_n.
2. Проміжні вузли шляху від вузла i до вузла j визначаємо по матриці S_n. Нехай s_{ij} = k, тоді маємо шлях $i \to k \to j$. Якщо далі s_{ik}=k і s_{kj}=j, тоді вважаємо, що увесь шлях визначений, оскільки знайдені усі проміжні вузли. Інакше повторюємо описану процедуру для шляхів від вузла i до вузла k і від вузла k до вузла j.

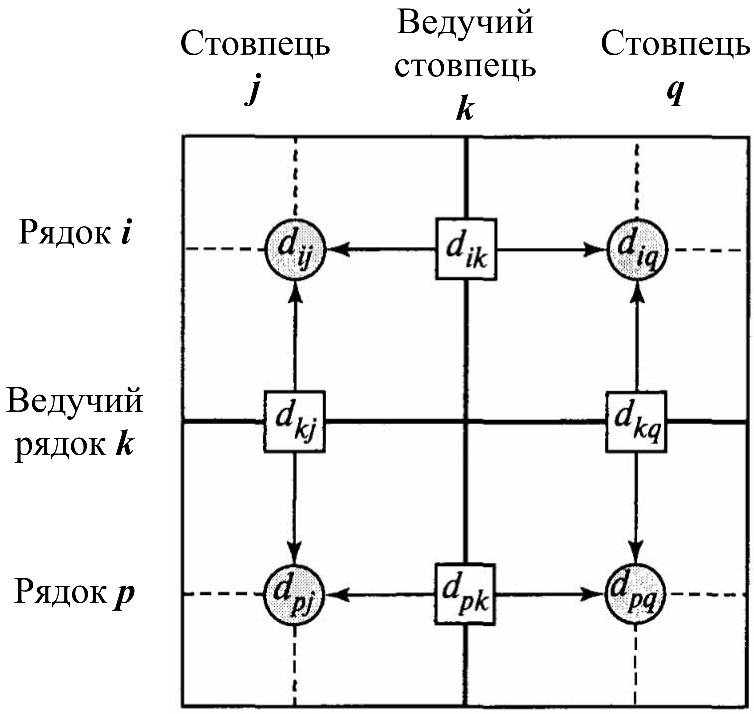


Рис. 4 – Представлення матриці D_{k-1}

Приклад 2

Знайдемо для мережі, показаної на рис. 5, найкоротші шляхи між будь-якими двома вузлами. Відстані між вузлами цієї мережі проставлені на малюнку біля відповідних ребер. Ребро (3, 5) орієнтовано, тому не допускається рух від вузла 5 до вузла 3. Усі інші ребра допускають рух в обидві сторони.

Етап 0. Початкові матриці D_0 і S_0 будуються безпосередньо за заданою схемою мережі. Матриця D_0 симетрична, за винятком пари елементів d_{35} і d_{53}, де $d_{53} = \infty$ (оскільки неможливий перехід від вузла 5 до вузла 3).

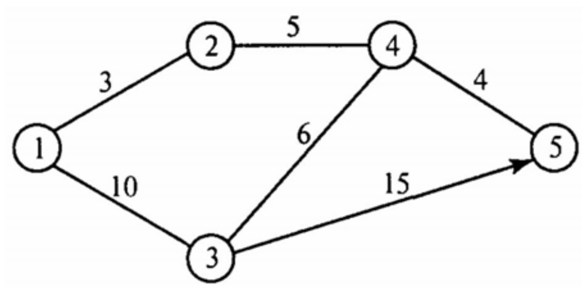


Рис. 5 – Початкова мережа

D_0	*1*	*2*	*3*	*4*	*5*
1	-	3	10	∞	∞
2	3	-	∞	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S_0	*1*	*2*	*3*	*4*	*5*
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Етап 1. У матриці D_0 виділені ведучі рядок і стовпець з номером (k=1). Затемненими представлені елементи d_{23} і d_{32} єдині серед елементів матриці D_0, значення яких можна поліпшити за допомогою трикутного оператора. Так, щоб на основі матриць D_0 і S_0 отримати матриці D_1 і S_1, виконуємо наступні дії.

1. Замінюємо d_{23} на $d_{21} + d_{13} = 3 + 10 = 13$ і встановлюємо $s_{23} = 1$.

2. Замінюємо d_{32} на $d_{31} + d_{12} = 10 + 3 = 13$ і встановлюємо $s_{32} = 1$. Матриці D_1 і S_1 мають наступний вигляд.

D_1	*1*	*2*	*3*	*4*	*5*
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S_1	*1*	*2*	*3*	*4*	*5*
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Етап 2. Вважаємо k = 2; у матриці D_1, виділені ведучі рядок і стовпець. Трикутний оператор застосовується до елементів матриць D_1 і S_1, виділеним затінюванням. В результаті отримуємо матриці D_2 і S_2.

D_2	*1*	*2*	*3*	*4*	*5*
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S_2	*1*	*2*	*3*	*4*	*5*
1	-	2	3	2	5
2	1	-	1	4	5
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Етап 3. Вважаємо k = 3; у матриці D_2 виділені ведучі рядок і стовпець. Трикутний оператор застосовується до затемнених елементів матриць D_2 і S_2. В результаті отримуємо матриці D_3 і S_3.

D_3	1	2	3	4	5
1	-	3	10	8	25
2	3	-	13	5	28
3	10	13	-	6	15
4	6	5	6	-	4
5	∞	∞	∞	4	-

S_3	1	2	3	4	5
1	-	2	3	2	3
2	1	-	1	4	3
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Етап 4. Вважаємо k = 4, ведучі рядок і стовпець в матриці D_3 виділені. Отримуємо нові матриці D_4 і S_4.

D_4	1	2	3	4	5
1	-	3	10	8	12
2	3	-	11	5	9
3	10	11	-	6	10
4	8	5	6	-	4
5	12	9	10	4	-

S_4	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

Етап 5. Вважаємо k = 5, ведучі рядок і стовпець в матриці D_4 виділені. Ніяких дій на цьому етапі не виконуємо; обчислення закінчені.

Кінцеві матриці D_4 і S_4 містять усю інформацію, необхідну для визначення найкоротших шляхів між будь-якими двома вузлами мережі. Наприклад, найкоротша відстань між вузлами 1 і 5 дорінює d_{15} =12.

Для визначення відповідних маршрутів нагадаємо, що сегмент маршруту (*i,j*) складається з ребра (*i,j*) тільки тоді, коли $s_{ij} = j$. Інакше вузли *i* і j пов'язані, принаймні, через один проміжний вузол. Наприклад, оскільки s_{15}= 4 і s_{45} = 5, спочатку найкоротший маршрут між вузлами 1 і 5 матиме вигляд 1→ 4→ 5. Але оскільки $s_{14} \neq 4$, вузли 1 і 4 у визначуваному шляху не пов'язані одним ребром (але в початковій мережі вони можуть бути пов'язані безпосередньо). Далі слід визначити проміжний вузол (вузли) між першим і четвертим вузлами. Маємо s_{14} =2 і s_{24} = 4, тому маршрут 1→ 4 замінюємо 1→ 2→ 4. Оскільки s_{12} = 2 і s_{24} = 4, інших проміжних вузлів немає. Комбінуючи певні сегменти маршруту, остаточно отримуємо наступний найкоротший шлях від вузла 1 до вузла 5: 1→ 2→ 4→ 5. Довжина цього шляху дорівнює 12 км.

ЗАВДАННЯ НА САМОСТІЙНУ РОБОТУ

Завдання 3. Знайдіть мінімальне остовне дерево для мережі, матриця «инциденций» якої відповідає номеру варіанту.

Завдання 4. Застосуєте алгоритм Флойда до мережі, матриця «інциденцій» якій відповідає номеру варіанту. Одне або декілька ребер можна зробити орієнтованими (необов'язково). Визначте найкоротші шляхи між парами вузлів: 1) першим і останнім; 2) другим і передостаннім.

Примітка. Відсутність шляху від одного вузла графа до іншого вузла позначається знаком "~" (хвиля). Якщо ребро є орієнтованим, то з одного боку матриці стоїть довжина шляху, а з іншого симетричного боку стоїть хвиля.

ВАРІАНТИ ПОЧАТКОВИХ ДАНИХ (теми 5, 6)

Початкові дані до завдань 3, 4																	
#1						#8						#15					
-	~	6	11	9	~	-	21	14	16	8	12	-	17	12	17	10	13
~	-	24	24	15	~	21	-	2	2	18	20	17	-	5	9	2	9
6	24	-	22	23	11	14	2	-	6	17	6	12	5	-	11	3	4
11	24	22	-	21	16	16	2	6	-	8	4	17	9	11	-	~	~
9	15	23	21	-	15	8	18	17	8	-	19	10	2	3	~	-	19
~	~	11	16	15	-	12	20	6	4	19	-	13	9	4	~	19	-
#2						#9						#16					
-	13	20	20	20	~	-	23	~	20	23	3	-	22	8	8	17	~
13	-	19	20	6	20	23	-	16	4	~	13	22	-	12	15	9	4
20	19	-	10	~	13	~	16	-	11	~	16	8	12	-	11	18	20
20	20	10	-	23	3	20	4	11	-	3	6	8	15	11	-	6	11
20	6	~	23	-	15	23	~	~	3	-	6	17	9	18	6	-	2
~	20	13	3	15	-	3	13	16	6	6	-	~	4	20	11	2	-
#3						#10						#17					
-	12	4	11	15	2	-	9	12	~	18	~	-	6	22	~	11	4
12	-	18	17	19	15	9	-	15	24	9	2	6	-	18	15	5	3
4	18	-	16	10	20	12	15	-	22	~	22	22	18	-	10	17	22
11	17	16	-	15	16	~	24	22	-	19	23	~	15	10	-	16	19
15	19	10	15	-	24	18	9	~	19	-	~	11	5	17	16	-	16
2	15	20	16	24	-	~	2	22	23	~	-	4	3	22	19	16	-
#4						#11						#18					
-	19	22	5	2	14	-	4	12	20	~	14	-	~	20	15	6	12
19	-	18	~	20	11	4	-	7	4	8	17	~	-	13	18	5	21
22	18	-	2	23	~	12	7	-	~	19	21	20	13	-	9	4	8
5	~	2	-	14	23	20	4	~	-	3	19	15	18	9	-	3	2
2	20	23	14	-	15	~	8	19	3	-	~	6	5	4	3	-	6
14	11	~	23	15	-	14	17	21	19	~	-	12	21	8	2	6	-
#5						#12						#19					
-	9	24	13	15	24	-	~	14	4	5	~	-	~	13	17	6	18
9	-	~	22	2	22	~	-	3	12	17	2	~	-	~	9	18	16
24	~	-	23	~	21	14	3	-	15	17	18	13	~	-	19	16	10
13	22	23	-	9	24	4	12	15	-	13	12	17	9	19	-	19	14
15	2	~	9	-	22	5	17	17	13	-	11	6	18	16	19	-	12
24	22	21	24	22	-	~	2	18	12	11	-	18	16	10	14	12	-
#6						#13						#20					
-	4	21	20	~	16	-	4	20	~	13	~	-	~	6	~	21	14
4	-	12	6	3	15	4	-	4	8	14	8	~	-	5	10	2	2
21	12	-	13	5	~	20	4	-	8	22	8	6	5	-	2	7	~
20	6	13	-	~	21	~	8	8	-	11	19	~	10	2	-	19	2
~	3	5	~	-	5	13	14	22	11	-	11	21	2	7	19	-	20
16	15	~	21	5	-	~	8	8	19	11	-	14	2	~	2	20	-
#7						#14											
-	~	15	~	~	~	-	15	11	15	2	~						
~	-	~	17	5	3	15	-	11	~	~	24						
15	~	-	11	4	14	11	11	-	20	21	20						
~	17	11	-	~	19	15	~	20	-	16	8						
~	5	4	~	-	21	2	~	21	16	-	5						
~	3	14	19	21	-	~	24	20	8	5	-						

ВАРІАНТИ ПОЧАТКОВИХ ДАНИХ (теми 5, 6)

Початкові дані до завдань 3, 4																	
#21						#28						#35					
-	8	17	11	4	8	-	15	17	17	20	4	-	2	14	12	5	23
8	-	4	17	5	21	15	-	~	19	23	13	2	-	~	18	5	~
17	4	-	18	18	18	17	~	-	22	12	14	14	~	-	12	14	2
11	17	18	-	8	16	17	19	22	-	13	7	12	18	12	-	9	9
4	5	18	8	-	12	20	23	12	13	-	9	5	5	14	9	-	10
8	21	18	16	12	-	4	13	14	7	9	-	23	~	2	9	10	-
#22						#29						#36					
-	12	17	18	~	23	-	21	~	22	24	~	-	~	16	21	14	18
12	-	20	3	~	17	21	-	8	10	13	16	~	-	2	~	9	7
17	20	-	24	~	12	~	8	-	15	9	10	16	2	-	4	2	5
18	3	24	-	20	8	22	10	15	-	14	~	21	~	4	-	5	4
~	~	~	20	-	23	24	13	9	14	-	5	14	9	2	5	-	~
23	17	12	8	23	-	~	16	10	~	5	-	18	7	5	4	~	-
#23						#30						#37					
-	19	8	22	19	7	-	18	22	24	24	13	-	12	19	~	5	18
19	-	14	5	7	~	18	-	10	2	10	7	12	-	15	5	~	7
8	14	-	8	14	~	22	10	-	13	4	15	19	15	-	21	24	4
22	5	8	-	~	8	24	2	13	-	9	2	~	5	21	-	8	8
19	7	14	~	-	19	24	10	4	9	-	5	5	~	24	8	-	~
7	~	~	8	19	-	13	7	15	2	5	-	18	7	4	8	~	-
#24						#31						#38					
-	~	6	5	22	13	-	19	3	12	11	23	-	12	16	~	4	9
~	-	6	22	3	24	19	-	5	19	16	23	12	-	~	18	22	3
6	6	-	2	14	14	3	5	-	23	23	~	16	~	-	4	13	8
5	22	2	-	2	~	12	19	23	-	13	7	~	18	4	-	17	23
22	3	14	2	-	2	11	16	23	13	-	~	4	22	13	17	-	6
13	24	14	~	2	-	23	23	~	7	~	-	9	3	8	23	6	-
#25						#32						#39					
-	19	~	7	~	17	-	7	7	24	23	15	-	23	18	14	11	18
19	-	7	12	23	2	7	-	8	23	22	24	23	-	11	24	20	9
~	7	-	13	~	~	7	8	-	13	19	8	18	11	-	~	19	10
7	12	13	-	22	14	24	23	13	-	~	3	14	24	~	-	11	22
~	23	~	22	-	~	23	22	19	~	-	7	11	20	19	11	-	20
17	2	~	14	~	-	15	24	8	3	7	-	18	9	10	22	20	-
#26						#33						#40					
-	6	9	~	22	19	-	10	8	5	5	~	-	9	4	~	21	16
6	-	4	23	~	20	10	-	18	21	24	3	9	-	9	15	17	4
9	4	-	~	~	13	8	18	-	15	~	4	4	9	-	23	~	18
~	23	~	-	~	11	5	21	15	-	18	12	~	15	23	-	9	~
22	~	~	~	-	~	5	24	~	18	-	23	21	17	~	9	-	23
19	20	13	11	~	-	~	3	4	12	23	-	16	4	18	~	23	-
#27						#34											
-	19	23	11	2	14	-	14	~	14	15	14						
19	-	3	24	7	24	14	-	11	18	20	17						
23	3	-	20	22	13	~	11	-	~	~	8						
11	24	20	-	5	11	14	18	~	-	11	11						
2	7	22	5	-	21	15	20	~	11	-	2						
14	24	13	11	21	-	14	17	8	11	2	-						

ТЕМИ 7, 8. ДЕТЕРМІНОВАНІ МОДЕЛІ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

Динамічне програмування (ДП) визначає оптимальне рішення *n*-мірної задачі шляхом її декомпозиції на *n* етапів, кожен з яких представляє підзадачу відносно однієї змінної. Обчислювальна перевага такого підходу полягає в тому, що ми займаємося рішенням одновимірних оптимізаційних підзадач замість великої n-мірної задачі. Фундаментальним принципом ДП, що становить основу декомпозиції задачі, є оптимальність. Оскільки природа кожного етапу рішення залежить від конкретного оптимізаційного завдання, ДП не пропонує обчислювальних алгоритмів безпосередньо для кожного етапу. Обчислювальні аспекти рішення оптимізаційних підзадач на кожному етапі проектуються і реалізуються окремо.

Обчислення в ДП виконуються рекурентно в тому сенсі, що оптимальне рішення однієї підзадачі використовується в якості початкових даних для наступної. Вирішивши останню підзадачу, ми отримаємо оптимальне рішення початкової задачі. Спосіб виконання рекурентних обчислень залежить від того, як проводиться декомпозиція початкового завдання. Зокрема, підзадачі зазвичай пов'язані між собою деякими загальними обмеженнями. Якщо здійснюється перехід від однієї підзадачі до іншої, то повинні враховуватися ці обмеження.

Рекурентний алгоритм прямого прогону

Приклад 1

Припустимо, необхідно вибрати найкоротший шлях між двома містами. Мережа доріг, показана на рис. 1, представляє можливі маршрути між початковим містом, що знаходиться у вузлі 1, і кінцевим пунктом, який знаходиться у вузлі 7. Маршрути проходять через проміжні міста, позначені на мережі вузлами з номерами 2-6.

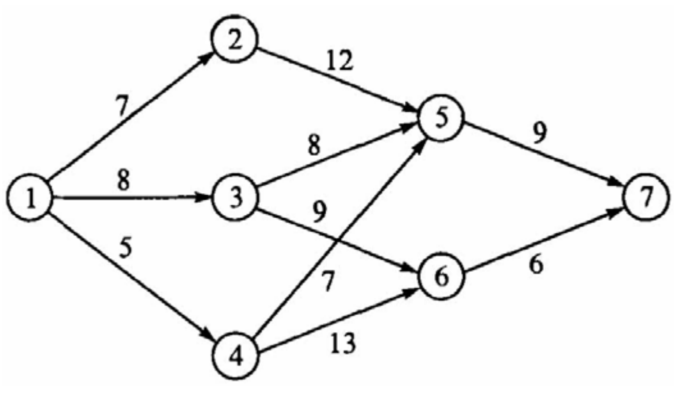


Рис. 1 – Початкова мережа

Ми можемо вирішити це завдання за допомогою повного перебору усіх маршрутів між вузлами 1 і 7 (є п'ять таких маршрутів). Проте у великій мережі повний перебір є неефективним з обчислювальної точки зору.

Щоб вирішити це завдання за допомогою методів динамічного програмування, спочатку розділимо її на етапи. Вертикальні пунктирні лінії на рис. 2 окреслюють три етапи завдання. Далі виконуються обчислення для кожного етапу окремо.

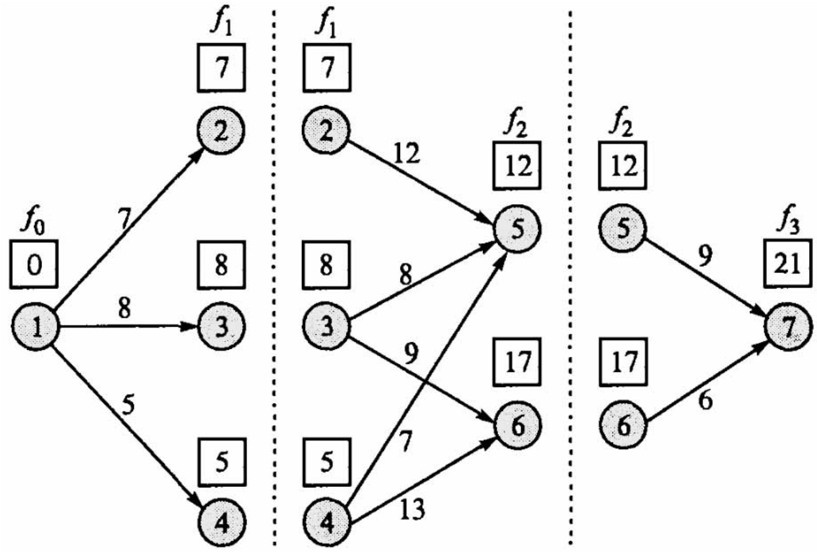


Рис. 2 – Етапи завдання

Загальне завдання полягає в обчисленні найкоротших (поступово накопичуваних) відстаней до усіх вершин етапу з наступним використанням цих відстаней в якості початкових даних для наступного етапу. Розглядаючи вузли, що відносяться до першого етапу, помічаємо, що кожен з вузлів 2, 3 і 4 пов'язаний з початковим вузлом 1 єдиною дугою (рис. 2). Отже, для першого етапу маємо наступне.

Етап 1. Підсумкові результати.
Найкоротший шлях з вузла 1 до вузла 2 дорівнює 7 км.
Найкоротший шлях з вузла 1 до вузла 3 дорівнює 8 км.
Найкоротший шлях з вузла 1 до вузла 4 дорівнює 5 км.

Далі переходимо до другого етапу для обчислення найкоротших (накопичених) відстаней до вузлів 5 і 6. Розглядаючи вузол 5 першим, з рис. 2 помічаємо, що є три можливі маршрути, по яких можна досягти вузла 5, а саме (2, 5), (3,5) і (4, 5). Ця інформація разом з найкоротшими відстанями до вузлів 2, 3, і 4 визначає найкоротшу (накопичену) відстань до вузла 5 таким чином.

$$\begin{pmatrix}\text{Найкоротша} \\ \text{відстань до вузла 5}\end{pmatrix} = \min_{i=2,3,4}\left\{\begin{pmatrix}\text{Найкоторша} \\ \text{відстань до вузла } i\end{pmatrix} + \begin{pmatrix}\text{Відстань від} \\ \text{вузла } i \text{ до вузла 5}\end{pmatrix}\right\} =$$

$$= \min \begin{Bmatrix} 7+12=19 \\ 8+8=16 \\ 5+7=12 \end{Bmatrix} = 12 \text{ (з вузла 4).}$$

Аналогічно для вузла 6 маємо наступне.

$$\begin{pmatrix} \text{Найкоротша} \\ \text{відстань до вузла 6} \end{pmatrix} = \min_{i=3,4} \left\{ \begin{pmatrix} \text{Найкоротша} \\ \text{відстань до вузла } i \end{pmatrix} + \begin{pmatrix} \text{Відстань від} \\ \text{вузла } i \text{ до вузла 6} \end{pmatrix} \right\} =$$

$$= \min \begin{Bmatrix} 8+9=17 \\ 5+13=18 \end{Bmatrix} = 17 \text{ (з вузла 3).}$$

Етап 2. Підсумкові результати.

Найкоротший шлях з вузла 4 до вузла 5 дорівнює 12 км.
Найкоротший шлях з вузла 3 до вузла 6 дорівнює 17 км.

Останнім кроком є третій етап. Кінцевий вузол 7 можна досягти як з вузла 5, так і 6. Використовуючи підсумкові результати етапу 2 і відстані від вузлів 5 і 6 до вузла 7, отримуємо наступне.

$$\begin{pmatrix} \text{Найкоротша} \\ \text{відстань до вузла 7} \end{pmatrix} = \min \begin{Bmatrix} 12+9=21 \\ 17+6=23 \end{Bmatrix} = 21 \text{ (з вузла 5).}$$

Етап 3. Підсумкові результати.

Найкоротший шлях до вузла 7 дорівнює 21 км (з вузла 5).

Приведені обчислення показують, що найкоротша відстань між вузлами 1 і 7 дорівнює 21 км. Міста, через які проходить найкоротший маршрут, визначаються таким чином. З підсумкових результатів третього етапу виходить, що вузол 7 зв'язується з вузлом 5. Далі з підсумкових результатів другого етапу виходить, що вузол 4 зв'язується з вузлом 5. Нарешті, з підсумкових результатів першого етапу виходить, що вузол 4 зв'язується з вузлом 1. Отже, оптимальним маршрутом є послідовність $1 \rightarrow 4 \rightarrow 5 \rightarrow 7$.

Формалізація рекурретного обчислення ДП

Нехай $f_i(x_i)$ — найкоротша відстань до вузла x_i на етапі i, $d(x_{i-1}, x_i)$ — відстань від вузла x_{i-1} до вузла x_i. Тоді f_i обчислюється на основі значень f_{i-1} за допомогою наступного рекурентного рівняння.

$$f_i(x_i) = \min_{\substack{\text{всі припустимі} \\ (x_{i-1}, x_i) \text{ маршрути}}} \{d(x_{i-1}, x_i) + f_{i-1}(x_{i-1})\}, \ i = 1,2,3$$

При i=1 вважаємо $f_0(x_0) \equiv 0$. Це рівняння показує, що найкоротші відстані $f_i(x_i)$ на етапі i мають бути виражені як функції наступного вузла x_i. У термінології динамічного програмування x_i іменується станом системи на етапі i.

Насправді стан системи на етапі i — це інформація, що зв'язує етапи між собою, при цьому оптимальні рішення для етапів, що залишилися, можуть прийматися без повторної перевірки того, як були отримані рішення на попередніх етапах. Таке визначення стану системи дозволяє розглядати кожен етап окремо і гарантує, що рішення є допустимим на кожному етапі.

Визначення стану системи призводить до наступного уніфікованого положення.

Принцип оптимальності. На кожному етапі оптимальна стратегія визначається незалежно від стратегій, використаних на попередніх етапах.

Застосування принципу оптимальності демонструється обчисленнями з прикладу 1. Наприклад, на етапі 3 ми використовуємо найкоротші шляхи до вузлів 5 і 6 і не цікавимося, як ці вузли були досягнуті з вузла 1.

Рекурентний алгоритм зворотного прогону

У прикладі 1 обчислення проводилися послідовно від першого етапу до третього. Така послідовність обчислень відома як алгоритм прямого прогону. Цей же приклад може бути вирішений за допомогою алгоритму зворотного прогону, відповідно до якого обчислення проводяться від третього етапу до першого.

Алгоритми прямого і зворотного прогону призводять до одного і того ж рішення. Незважаючи на те що алгоритм прямого прогону представляється логічнішим, в спеціальній літературі, присвяченій ДП, незмінно використовується алгоритм зворотного прогону. Причина цього в тому, що в загальному випадку алгоритм зворотного прогону може бути ефективнішим з обчислювальної точки зору. Продемонструємо використання алгоритму зворотного прогону на прикладі 1. Також представимо обчислення динамічного програмування в компактній табличній формі.

Приклад 2

Рекурентне рівняння для алгоритму зворотного прогону в прикладі 1 має вигляд

$$f_i(x_i) = \min_{\substack{\text{всі припустимі} \\ (x_{i-1}, x_i) \text{ маршрути}}} \{d(x_{i-1}, x_i) + f_{i-1}(x_{i-1})\}, \ i = 1,2,3$$

Де $f_4(x_4) \equiv 0$ для $x_4 = 7$. Відповідною послідовністю обчислень буде $f_3 \to f_2 \to f_1$.

Етап 3. Оскільки вузол 7 ($x_4 = 7$) пов'язаний з вузлами 5 і 6 {$x_3 = 5$ і 6) тільки одним маршрутом, альтернативи для вибору відсутні, а результати третього етапу можна підсумувати таким чином.

Етап 3	$d(x_3, x_4)$	Оптимальне рішення	
x_3	$x_4 = 7$	$f_3(x_3)$	x_4^*
5	9	9	7
6	6	6	7

Етап 2. Оскільки маршруту (2, 6) не існує, відповідна альтернатива не розглядається. Використовуючи значення $f_3(x_3)$ отримані на третьому етапі, ми можемо порівняти допустимі альтернативні рішення, як показано в наступній таблиці.

Етап 2	$d(x_2, x_3) + f_3(x_3)$		Оптимальне рішення	
x_2	$x_3 = 5$	$x_3 = 6$	$f_2(x_2)$	x_3^*
2	12+9=21	-	21	5
3	8+9=17	9+6=15	15	6
4	7+9=16	13+6=19	16	5

Оптимальне рішення другого етапу означає наступне. Якщо ви знаходитеся у вузлі 2 або 4, найкоротший шлях до вузла 7 проходить через вузол 5, а якщо у вузлі 3 — через вузол 6.

Етап 1. З вузла 1 починаються три альтернативні маршрути: (1, 2), (1, 3) і (1, 4). Використовуючи значення $f_2(x_2)$ отримані на другому етапі, обчислюємо дані наступної таблиці.

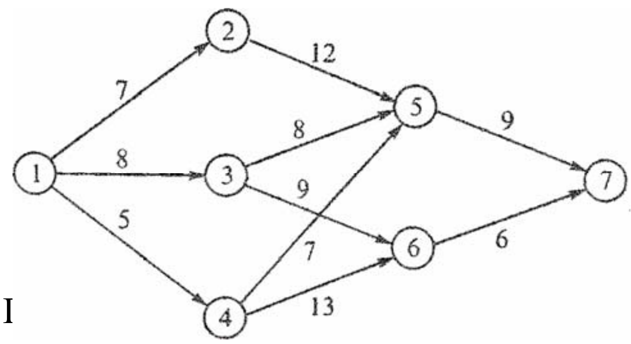
Оптимальне рішення на першому етапі показує, що найкоротший шлях проходить через вузол 4. Далі з оптимального рішення на другому етапі витікає, що з вузла 4 необхідно рухатися у вузол 5. Нарешті, з оптимального рішення на третьому етапі витікає, що вузол 5 пов'язаний з вузлом 7. Отже, повним маршрутом, що має найкоротшу довжину, є 1→ 4→ 5→ 7, і його довжина дорівнює 21 км.

Етап 1	$d(x_1, x_2) + f_2(x_2)$			Оптимальне рішення	
x_1	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$f_1(x_1)$	x_2^*
1	7+21=28	8+15=23	5+16=21	21	4

ЗАВДАННЯ НА САМОСТІЙНУ РОБОТУ

Завдання 1. Для мережі, матриця «інциденцій» якій відповідає номеру варіанту, знайдіть найкоротший шлях між першим і останнім вузлами. Для вирішення застосуєте рекурентний алгоритм прямого прогону ДП. Довжини маршрутів приведені у відповідній матриці «інциденцій»; наприклад, для завдання з прикладу 1 мережа і її матриця инциденций виглядають таким чином:

d	*1*	*2*	*3*	*4*	*5*	*6*	*7*
1	-	7	8	5	~	~	~
2	~	-	~	~	12	~	~
3	~	~	-	~	8	9	~
4	~	~	~	-	7	13	~
5	~	~	~	~	-	~	9
6	~	~	~	~	~	-	6
7	~	~	~	~	~	~	-



Примітка. Тут, відсутність шляху від одного вузла графа до іншого вузла позначається знаком "~". Якщо ребро є орієнтованим, то з одного боку матриці стоїть довжина шляху, а з іншого симетричного боку стоїть "~".

Завдання 2. Для задачі із завдання 1 отримаєте рекурентне співвідношення зворотного прогону і використовуйте його для отримання оптимального рішення.

ВАРІАНТИ ПОЧАТКОВИХ ДАНИХ (ТЕМИ 7,8)

Початкові дані до завдань 1, 2

#1

-	7	23	~	~	~	~	~	~
~	-	~	19	4	15	~	~	~
~	~	-	2	14	16	~	~	~
~	~	~	-	~	~	3	19	~
~	~	~	~	-	~	21	15	~
~	~	~	~	~	-	23	10	~
~	~	~	~	~	~	-	~	24
~	~	~	~	~	~	~	-	6
~	~	~	~	~	~	~	~	-

#2

-	19	20	~	~	~	~	~	~
~	-	~	5	22	14	~	~	~
~	~	-	24	11	18	~	~	~
~	~	~	-	~	~	3	13	~
~	~	~	~	-	~	14	7	~
~	~	~	~	~	-	21	15	~
~	~	~	~	~	~	-	~	9
~	~	~	~	~	~	~	-	21
~	~	~	~	~	~	~	~	-

#3

-	6	22	~	~	~	~	~	~
~	-	~	12	17	8	~	~	~
~	~	-	6	7	5	~	~	~
~	~	~	-	~	~	24	24	~
~	~	~	~	-	~	4	7	~
~	~	~	~	~	-	7	5	~
~	~	~	~	~	~	-	~	21
~	~	~	~	~	~	~	-	23
~	~	~	~	~	~	~	~	-

#4

-	8	15	~	~	~	~	~	~
~	-	~	13	2	7	~	~	~
~	~	-	18	18	18	~	~	~
~	~	~	-	~	~	11	5	~
~	~	~	~	-	~	16	14	~
~	~	~	~	~	-	9	3	~
~	~	~	~	~	~	-	~	16
~	~	~	~	~	~	~	-	12
~	~	~	~	~	~	~	~	-

#5

-	21	24	~	~	~	~	~	~
~	-	~	23	5	8	~	~	~
~	~	-	10	10	2	~	~	~
~	~	~	-	~	~	17	24	~
~	~	~	~	-	~	22	14	~
~	~	~	~	~	-	11	24	~
~	~	~	~	~	~	-	~	12
~	~	~	~	~	~	~	-	23
~	~	~	~	~	~	~	~	-

#6

-	7	9	7	~	~	~	~
~	-	~	~	11	6	21	~
~	~	-	~	23	6	24	~
~	~	~	-	13	18	17	~
~	~	~	~	-	~	~	3
~	~	~	~	~	-	~	9
~	~	~	~	~	~	-	7
~	~	~	~	~	~	~	-

#7

-	14	13	15	~	~	~	~
~	-	~	~	20	10	14	~
~	~	-	~	14	4	11	~
~	~	~	-	2	19	13	~
~	~	~	~	-	~	~	14
~	~	~	~	~	-	~	17
~	~	~	~	~	~	-	5
~	~	~	~	~	~	~	-

#8

-	19	24	11	~	~	~	~
~	-	~	~	5	22	12	~
~	~	-	~	7	11	15	~
~	~	~	-	17	22	4	~
~	~	~	~	-	~	~	23
~	~	~	~	~	-	~	21
~	~	~	~	~	~	-	6
~	~	~	~	~	~	~	-

#9

-	8	17	12	~	~	~	~
~	-	~	~	21	19	10	~
~	~	-	~	4	18	13	~
~	~	~	-	24	16	13	~
~	~	~	~	-	~	~	14
~	~	~	~	~	-	~	18
~	~	~	~	~	~	-	20
~	~	~	~	~	~	~	-

#10

-	23	9	5	~	~	~	~
~	-	~	~	5	23	24	~
~	~	-	~	11	24	3	~
~	~	~	-	21	3	20	~
~	~	~	~	-	~	~	9
~	~	~	~	~	-	~	10
~	~	~	~	~	~	-	2
~	~	~	~	~	~	~	-

ВАРІАНТИ ПОЧАТКОВИХ ДАНИХ (ТЕМИ 7,8)

Початкові дані до завдань 1, 2																	
#11									#12								
-	3	20	~	~	~	~	~	~	-	7	8	~	~	~	~	~	~
~	-	~	11	15	~	~	~	~	~	-	~	23	14	~	~	~	~
~	~	-	17	22	~	~	~	~	~	~	-	16	14	~	~	~	~
~	~	~	-	~	6	7	21	~	~	~	~	-	~	23	16	20	~
~	~	~	~	-	6	10	3	~	~	~	~	~	-	14	20	4	~
~	~	~	~	~	-	~	~	15	~	~	~	~	~	-	~	~	17
~	~	~	~	~	~	-	~	22	~	~	~	~	~	~	-	~	7
~	~	~	~	~	~	~	-	17	~	~	~	~	~	~	~	-	9
~	~	~	~	~	~	~	~	-	~	~	~	~	~	~	~	~	-
#13									#14								
-	7	4	~	~	~	~	~	~	-	13	5	~	~	~	~	~	~
~	-	~	11	16	~	~	~	~	~	-	~	13	12	~	~	~	~
~	~	-	2	14	~	~	~	~	~	~	-	7	16	~	~	~	~
~	~	~	-	~	18	21	3	~	~	~	~	-	~	4	17	14	~
~	~	~	~	-	24	17	18	~	~	~	~	~	-	5	18	18	~
~	~	~	~	~	-	~	~	10	~	~	~	~	~	-	~	~	18
~	~	~	~	~	~	-	~	4	~	~	~	~	~	~	-	~	13
~	~	~	~	~	~	~	-	13	~	~	~	~	~	~	~	-	7
~	~	~	~	~	~	~	~	-	~	~	~	~	~	~	~	~	-
#15									#16								
-	9	12	~	~	~	~	~	~	-	20	10	7	19	~	~	~	~
~	-	~	21	6	~	~	~	~	~	-	~	~	~	23	9	7	~
~	~	-	21	18	~	~	~	~	~	~	-	~	~	18	17	11	~
~	~	~	-	~	4	15	20	~	~	~	~	-	~	19	19	12	~
~	~	~	~	-	14	13	3	~	~	~	~	~	-	14	12	5	~
~	~	~	~	~	-	~	~	12	~	~	~	~	~	-	~	~	20
~	~	~	~	~	~	-	~	18	~	~	~	~	~	~	-	~	20
~	~	~	~	~	~	~	-	8	~	~	~	~	~	~	~	-	14
~	~	~	~	~	~	~	~	-	~	~	~	~	~	~	~	~	-
#17									#18								
-	3	22	3	15	~	~	~	~	-	14	3	18	14	~	~	~	~
~	-	~	~	~	4	21	2	~	~	-	~	~	~	13	11	17	~
~	~	-	~	~	22	23	11	~	~	~	-	~	~	8	22	22	~
~	~	~	-	~	23	5	20	~	~	~	~	-	~	9	16	4	~
~	~	~	~	-	23	11	6	~	~	~	~	~	-	8	19	19	~
~	~	~	~	~	-	~	~	18	~	~	~	~	~	-	~	~	11
~	~	~	~	~	~	-	~	3	~	~	~	~	~	~	-	~	15
~	~	~	~	~	~	~	-	20	~	~	~	~	~	~	~	-	3
~	~	~	~	~	~	~	~	-	~	~	~	~	~	~	~	~	-
#19									#20								
-	20	2	23	19	~	~	~	~	-	21	4	23	23	~	~	~	~
~	-	~	~	~	7	17	22	~	~	-	~	~	~	7	13	4	~
~	~	-	~	~	2	4	4	~	~	~	-	~	~	8	18	19	~
~	~	~	-	~	19	5	9	~	~	~	~	-	~	12	24	16	~
~	~	~	~	-	18	7	2	~	~	~	~	~	-	24	8	14	~
~	~	~	~	~	-	~	~	24	~	~	~	~	~	-	~	~	3
~	~	~	~	~	~	-	~	18	~	~	~	~	~	~	-	~	12
~	~	~	~	~	~	~	-	14	~	~	~	~	~	~	~	-	22
~	~	~	~	~	~	~	~	-	~	~	~	~	~	~	~	~	-

ВАРІАНТИ ПОЧАТКОВИХ ДАНИХ (ТЕМИ 7,8)

Початкові дані до завдань 1, 2

#21

-	12	23	12	~	~	~	~	~
~	-	~	~	9	15	22	19	~
~	~	-	~	22	12	16	15	~
~	~	~	-	4	24	13	8	~
~	~	~	~	-	~	~	~	8
~	~	~	~	~	-	~	~	10
~	~	~	~	~	~	-	~	13
~	~	~	~	~	~	~	-	16
~	~	~	~	~	~	~	~	-

#22

-	21	19	4	~	~	~	~	~
~	-	~	~	12	19	22	4	~
~	~	-	~	24	3	12	8	~
~	~	~	-	16	2	21	21	~
~	~	~	~	-	~	~	~	6
~	~	~	~	~	-	~	~	14
~	~	~	~	~	~	-	~	18
~	~	~	~	~	~	~	-	9
~	~	~	~	~	~	~	~	-

#23

-	14	3	6	~	~	~	~	~
~	-	~	~	5	13	14	14	~
~	~	-	~	4	12	12	5	~
~	~	~	-	5	19	12	16	~
~	~	~	~	-	~	~	~	5
~	~	~	~	~	-	~	~	5
~	~	~	~	~	~	-	~	2
~	~	~	~	~	~	~	-	12
~	~	~	~	~	~	~	~	-

#24

-	7	11	5	~	~	~	~	~
~	-	~	~	4	24	14	8	~
~	~	-	~	15	21	17	20	~
~	~	~	-	24	6	21	4	~
~	~	~	~	-	~	~	~	8
~	~	~	~	~	-	~	~	7
~	~	~	~	~	~	-	~	19
~	~	~	~	~	~	~	-	13
~	~	~	~	~	~	~	~	-

#25

-	3	5	8	~	~	~	~	~
~	-	~	~	23	4	19	5	~
~	~	-	~	8	17	11	17	~
~	~	~	-	7	10	19	8	~
~	~	~	~	-	~	~	~	5
~	~	~	~	~	-	~	~	24
~	~	~	~	~	~	-	~	13
~	~	~	~	~	~	~	-	11
~	~	~	~	~	~	~	~	-

#26

-	7	12	16	~	~	~	~	~
~	-	~	~	8	5	~	~	~
~	~	-	~	20	4	~	~	~
~	~	~	-	24	8	~	~	~
~	~	~	~	-	~	21	9	~
~	~	~	~	~	-	12	13	~
~	~	~	~	~	~	-	~	13
~	~	~	~	~	~	~	-	13
~	~	~	~	~	~	~	~	-

#27

-	4	10	24	~	~	~	~	~
~	-	~	~	5	16	~	~	~
~	~	-	~	10	18	~	~	~
~	~	~	-	11	22	~	~	~
~	~	~	~	-	~	8	21	~
~	~	~	~	~	-	4	22	~
~	~	~	~	~	~	-	~	6
~	~	~	~	~	~	~	-	13
~	~	~	~	~	~	~	~	-

#28

-	20	13	5	~	~	~	~	~
~	-	~	~	9	24	~	~	~
~	~	-	~	21	23	~	~	~
~	~	~	-	12	7	~	~	~
~	~	~	~	-	~	7	14	~
~	~	~	~	~	-	4	19	~
~	~	~	~	~	~	-	~	11
~	~	~	~	~	~	~	-	10
~	~	~	~	~	~	~	~	-

#29

-	21	8	18	~	~	~	~	~
~	-	~	~	13	24	~	~	~
~	~	-	~	5	5	~	~	~
~	~	~	-	21	9	~	~	~
~	~	~	~	-	~	6	15	~
~	~	~	~	~	-	19	5	~
~	~	~	~	~	~	-	~	2
~	~	~	~	~	~	~	-	14
~	~	~	~	~	~	~	~	-

#30

-	5	8	8	~	~	~	~	~
~	-	~	~	4	2	~	~	~
~	~	-	~	14	10	~	~	~
~	~	~	-	14	4	~	~	~
~	~	~	~	-	~	23	5	~
~	~	~	~	~	-	23	12	~
~	~	~	~	~	~	-	~	14
~	~	~	~	~	~	~	-	23
~	~	~	~	~	~	~	~	-

СПИСОК ЛІТЕРАТУРИ

1. Ладанюк, А. П. Основи системного аналізу : Навчальний посібник / А. П. Ладанюк. – Вінниця : Нова книга, 2004. – 176 с.
2. Сорока, К. О. Основи теорії систем і системного аналізу : Навч. посібник / К. О. Сорока. – 2-ге вид., перероб. та випр. – Х. : Тимченко А.М., 2005. – 288 с.
3. Катренко, А. В. Системний аналіз : Підручник / А. В. Катренко ; За наук. ред. В. В. Пасічника. – Львів : «Новий світ-2000», 2011. – 396 с.
4. Катренко, А. В. Системний аналіз об'єктів та процесів комп'ютеризації : Навчальний посібник / А. В. Катренко. – Львів : «Новий світ-2000», 2003. – 424 с.
5. Згуровський, М. З. Основи системного аналізу : Підручник / М. З. Згуровський, Н. Д. Панкратова ; За заг. ред. М. З. Згуровського. – К. : Видавнича група BHV, 2007. – 544 с.
6. Задорский, В. М. Теория технических систем : Учебное пособие / В. М. Задорский. – Днепролпетровск : ГВУЗ УГХТУ, 2016. – 442 с.
7. Шамровський, О. Д. Системний аналіз: математичні методи та застосування : Навчальний посібник / О. Д. Шамровський. – Львів : Магнолія 2006, 2010. – 275 с.
8. Тимченко, А. А. Основи системного проектування та системного аналізу складних об'єктів : Основи системного підходу та системного аналізу об'єктів нової техніки. Навчальний посібник / А. А. Тимченко ; За ред. Ю. Г. Леги. – К. : Либідь, 2004. – 288 с.
9. Писарук, Н. Н. Исследование операций / Н. Н. Писарук. – Минск : БГУ, 2012. – 281 с.
10. Теоретические основы системного анализа / Новосельцев В. И. [и др.] ; под ред. В. И. Новосельцева. – М. : Майор, 2006. – 592 с. : ил.
11. Коломоец, Ф. Г. Основы системного анализа и теории принятия решений: пособие для исследователей, управленцев и студентов вузов / Ф.Г. Коломоец. – Мн. : Тесей, 2006. – 320 с.
12. Катренко, А. В. Дослідження операцій. Підручник. – Львів : “Магнолія Плюс”, 2004. – 549 с.
13. Карагодова, О.О. Дослідження операцій : Навч. посібник. / О. О. Карагодова, В. Р. Кігель, В. Д. Рожок. – К. : Центр учбової літератури, 2007. – 256 с.
14. Горбань, О.М. Основи теорії систем і системного аналізу : Навчальний посібник. / О.М. Горбань, В.Є. Бахрушин. – Запоріжжя : ГУ “ЗІДМУ”, 2004. – 204 с.
15. Таха, Хемди А. Введение в исследование операций, 7-е издание. : Пер. с англ. – М. : Издательский дом “Вильямс”, 2005. – 912 с. : ил. – Парал. тит. англ.
16. Hillier, F. S., and G. J. Lieberman : Introduction to operations research, 10th ed., McGraw-Hill Education, New York, 2015
17. Сурмин, Ю. П. Теория систем и системный анализ : Учеб. пособие. – К. :

МАУП, 2003. – 368 с.