

## **Практична робота 1. Моделі обчислювальних процесів**

**Мета:** Вивчення моделей обчислювальних процесів

### **Теоретичні відомості**

Модель - це фізична або інформаційна система, що представляє собою об'єкт дослідження адекватно цілям дослідження.

Фізичні моделі утворюються із сукупності матеріальних об'єктів. Для їх побудови використовуються різні властивості, причому природа застосовуваних в моделі матеріальних елементів не обов'язково та ж, що і в досліджуваному об'єкті. Прикладом фізичної моделі є макет.

Абстрактна модель - це опис об'єкта дослідження на деякій мові. Абстрактність моделі виявляється в тому, що її компонентами є поняття, а не фізичні елементи, (наприклад: словесні описи, креслення, схеми, графіки, таблиці, програми, алгоритми, математичні описи).

Необхідна умова для переходу від дослідження об'єкта до дослідження моделі та подальшого перенесення його результатів на об'єкт дослідження - вимога адекватності моделі та об'єкта. Адекватність передбачає відтворення моделлю з необхідною повнотою всіх властивостей об'єкта, істотних для цілей даного дослідження.

Трудомісткість алгоритмів - кількість обчислювальної роботи необхідної для реалізації алгоритму.

Складність алгоритму - мінімальна кількість інформації, необхідної для його опису. Зазвичай в практиці складність алгоритму визначається довжиною запису алгоритму в термінах певної алгоритмічної системи. Наприклад, складність алгоритму можна характеризувати числом операторів в програмі або числом команд програми в машинному коді.

Складність завдання складається з складності алгоритму та кількості даних. Кількість даних, що відносяться до задачі, характеризується числом байтів, за допомогою яких представляються дані. Маючи в своєму

розпорядженні відомості про складність алгоритму і кількості даних, можна визначити потребу завдання в ресурсах пам'яті.

Якщо складність алгоритму характеризує потребу алгоритму в пам'яті, то трудомісткість - його потреба в часі, пов'язаному з періодом роботи сукупності пристроїв, засобами яких реалізується алгоритм. Трудомісткість алгоритму тому, іноді називають складністю обчислень. Оцінюється трудомісткість алгоритму кількістю операцій, виконуваних з метою обробки, введення і виведення інформації в процесі виконання завдання. Кожній реалізації алгоритму властивий елемент випадковості, пов'язаний з тим, що вихідні дані представляють собою в загальному випадку випадкову вибірку з безлічі вихідних даних, до яких застосуємо алгоритм. Тому повна характеристика трудомісткості передбачає опис кількості операцій, які виконуються за одну реалізацію алгоритму, випадковими величинами, тобто передбачає визначення законів розподілу числа операцій у реалізації алгоритму. Отримання таких відомостей про алгоритм - складний і тривалий процес. У зв'язку з цим трудомісткість зазвичай характеризують наближено, наприклад, тільки математичними очікуваннями числа виконуваних операцій.

У першому наближенні трудомісткість алгоритму можна охарактеризувати такою сукупністю параметрів:

$V$  - середня кількість процесорних операцій, виконуваних за одну реалізацію алгоритму (при одному прогоні програми);

$N_1, N_2, \dots, N_n$  - середня кількість звернень до файлів  $F_1, F_2, \dots, F_n$  відповідно за одну реалізацію алгоритму;

$Q_1, Q_2, \dots, Q_n$  - середня кількість інформації (байтів) передаються за одне звернення до файлів  $F_1, F_2, \dots, F_n$  відповідно.

Всі оператори алгоритму підрозділяються на функціональні, переходу, вводу/ виводу.

Функціональний оператор - задає сукупність обчислювальних операцій. Оператор переходу - задає правило вибору одного з можливих шляхів розвитку

обчислювального процесу, відповідного поточним значенням даних, відносини між якими представляються предикатами.

Оператор введення / виведення - задає звернення до певного файлу з метою передачі деякої кількості даних.

Перші два типи операторів задають сукупність обчислювальних операцій над даними і відносяться до класу операторів, які називаються основними.

Сукупність операторів алгоритму та зв'язків між ними наочно представляються графом алгоритму, вершини якого відповідають операторам алгоритму, а дуги відображають зв'язки між операторами. Серед вершин графа виділяють початкову, кінцеву та операторні.

Номери вершин графа позначимо  $0, 1, \dots, k$ , де  $0$  - початкова,  $k$  - кінцева вершина графа. Номери  $1, 2, \dots, k-1$  ідентифікують оператори алгоритму. Таким чином, граф алгоритму дає наочне уявлення структури алгоритму, визначаючи множина операторів  $V = \{v_1, \dots, v_{k-1}\}$  та дуг  $D = \{(i, j)\}; i = 0, \dots, k-1; j = 1, \dots, k$ , зв'язуючих операторів.

Переходи між операторами  $v_i$  і  $v_j$  розглядаємо як випадкові події і характеризуємо можливостями  $p_{ij}$ , тобто кожна дуга  $(i, j)$  графа алгоритму позначається числом  $p_{ij}$ .

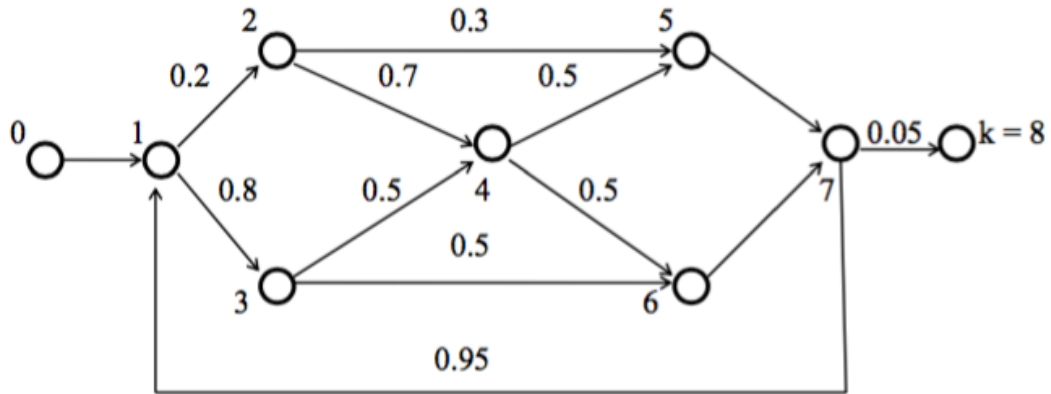
Середнє число  $n_1, \dots, n_{k-1}$  перебувань марковського процесу в неповоротних станах  $S_1, \dots, S_{k-1}$  (тобто шукане середнє число  $n_1, \dots, n_{k-1}$  звернень до операторів  $v_1, \dots, v_{k-1}$ ) визначається коренями системи лінійних алгебраїчних рівнянь:

$$\begin{cases} n_1 = 1 + p_{11}n_1 + p_{21}n_2 + \dots + p_{k-1,1}n_{k-1} \\ n_2 = p_{12}n_1 + p_{22}n_2 + \dots + p_{k-1,2}n_{k-1} \\ \dots \\ n_{k-1} = p_{1,k-1}n_1 + p_{2,k-1}n_2 + \dots + p_{k-1,k-1}n_{k-1} \end{cases}$$

Після перетворень отримаємо:

$$\begin{cases} (p_{11} - 1)n_1 + p_{21}n_2 + \dots + p_{k-1,1}n_{k-1} = -1 \\ p_{12}n_1 + (p_{22} - 1)n_2 + \dots + p_{k-1,2}n_{k-1} = 0 \\ \dots \\ p_{1,k-1}n_1 + p_{2,k-1}n_2 + \dots + (p_{k-1,k-1} - 1)n_{k-1} = 0 \end{cases}$$

Нехай для деякої конкретної реалізації граф алгоритму має вигляд:



Маємо систему з семи лінійних алгебраїчних рівнянь:

$$\begin{cases} -n_1 & & & & & & +0.95n_7 & = -1 \\ 0.2n_1 & -n_2 & & & & & & = 0 \\ 0.8n_1 & & -n_3 & & & & & = 0 \\ & 0.7n_2 & +0.5n_3 & -n_4 & & & & = 0 \\ & 0.3n_2 & & +0.5n_4 & -n_5 & & & = 0 \\ & & 0.5n_3 & +0.5n_4 & & -n_6 & & = 0 \\ & & & & n_5 & +n_6 & -n_7 & = 0 \end{cases}$$

Рішення системи рівнянь визначає середнє число влучень обчислювального процесу в стану  $S_1, \dots, S_7$ :

$$\begin{aligned} n_1 = 20, \quad n_3 = 16, \quad n_5 = 6.6, \quad n_7 = 20. \\ n_2 = 4, \quad n_4 = 10.8, \quad n_6 = 13.4, \end{aligned}$$

Нехай всі оператори алгоритму - основні, а кількість операцій -  $k_i$ , породжуваних оператором  $v_i$ , постійна і дорівнює 1. Тоді трудомісткість  $\theta_{\text{осн}} = \sum_{i=1}^7 k_i n_i = 20 + 4 + \dots + 20 = 90.8$  алгоритму буде дорівнює операцій.

Якщо, наприклад, оператори  $v_4$  і  $v_7$  є операторами введення/виведення, а кількість операцій, породжуваних кожним з основних операторів, так само  $k_1 = 500, k_2 = 500, k_3 = 50, k_5 = 300, k_6 = 20$ , відповідно, то трудомісткість по основним операторам складе величину  $\theta_{\text{осн}} = 500 * 20 + 500 * 4 + 50 * 16 + 300 * 6.6 + 20 * 13.4 = 15\,048$  операцій.

Нехай при зверненні з оператора  $v_4$  до файлу  $F_4$  передається  $l_4 = 1800$  байт даних, а при зверненні з оператора  $v_7$  до файлу  $F_7$  передається  $l_7 = 2540$  байт. Тоді

трудомісткість по вводу/виводу визначається, як  $\theta_{B/B} = \sum_{4,7} \theta_{B/B}^{(h)} = \theta_{B/B}^{(4)} + \theta_{B/B}^{(7)}$ . В

$$\theta_{B/B}^{(4)} = \sum_{v_4 \in S_4} n_i \ell_i = n_4 \ell_4 = 10.8 * 1800 = 19440 \text{ байт.}$$

свою чергу . Аналогічно

$$\theta_{B/B}^{(7)} = n_7 \ell_7 = 20 * 2540 = 50800 \text{ байт.} \quad \text{Тоді } \theta_{B/B} = 19440 + 50800 = 70240 \text{ байт.}$$

Хід роботи

1. Ознайомитись з матеріалами практичної роботи
2. Побудувати по таблиці 1.1, у відповідності з варіантом, граф алгоритму.
3. Побудувати математичну модель обчислювального процесу для оцінки трудоемності алгоритму.
4. Підготувати звіт по виконаній роботі.

Таблица 1

Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13	14
P <sub>12</sub>	0.2	0.1	1	0.5	1	1	0.3	1	0.2	0.3	0.1	1	1	0.2
P <sub>13</sub>	0.2	0.3		0.5			0.7		0.4	0.7	0.1			0.8
P <sub>14</sub>	0.6	0.6							0.4		0.8			
P <sub>23</sub>			0.1		0.3	0.6		0.2				0.2	0.5	
P <sub>24</sub>			0.3	0.4	0.7	0.4	0.5	0.3		0.7		0.8	0.5	0.3
P <sub>25</sub>	1	1	0.6	0.6			0.5	0.5	1	0.3	1			0.7
P <sub>34</sub>					1							1		
P <sub>35</sub>	0.1	0.3		0.3		0.1	0.1		0.1	0.6	0.2		0.2	0.2
P <sub>36</sub>	0.9	0.3	1			0.9	0.2	1	0.2		0.8		0.8	0.2
P <sub>37</sub>		0.4		0.7			0.7		0.7	0.4				0.6
P <sub>45</sub>					0.5							0.3		
P <sub>46</sub>	1		1	1	0.5	0.3	1	1		1	1	0.7	0.2	1
P <sub>47</sub>		1				0.7			1				0.8	
P <sub>56</sub>		1	1	1			1	1	1	1				1
P <sub>57</sub>	1				1	1					1	1	1	
P <sub>61</sub>			0.9					0.8						
P <sub>67</sub>	1	1	0.1		1	1	1	0.2	1		1	1	1	1
P <sub>68</sub>				1						1				
P <sub>71</sub>	0.9	0.8					0.8		0.8		0.9			0.9
P <sub>72</sub>					0.8							0.5		
P <sub>78</sub>				1		1				1			1	
P <sub>81</sub>				0.9		0.7				0.8			0.6	

Вариант	15	16	17	18	19	20	21	22	23	24	25	26	27	28
P <sub>12</sub>	0.2	0.1	1	0.5	1	1	0.3	1	0.2	0.3	0.1	1	1	0.2
P <sub>13</sub>	0.2	0.3		0.5			0.7		0.4	0.7	0.1			0.8
P <sub>14</sub>	0.6	0.6							0.4		0.8			
P <sub>23</sub>			0.1		0.3	0.6		0.2				0.2	0.5	
P <sub>24</sub>			0.3	0.4	0.7	0.4	0.5	0.3		0.7		0.8	0.5	0.3
P <sub>25</sub>	1	1	0.6	0.6			0.5	0.5	1	0.3	1			0.7
P <sub>34</sub>					1							1		
P <sub>35</sub>	0.1	0.3		0.3		0.1	0.1		0.1	0.6	0.2		0.2	0.2
P <sub>36</sub>	0.9	0.3	1			0.9	0.2	1	0.2		0.8		0.8	0.2
P <sub>37</sub>		0.4		0.7			0.7		0.7	0.4				0.6
P <sub>45</sub>					0.5							0.3		
P <sub>46</sub>	1		1	1	0.5	0.3	1	1		1	1	0.7	0.2	1
P <sub>47</sub>		1				0.7			1				0.8	
P <sub>56</sub>		1	1	1			1	1	1	1				1
P <sub>57</sub>	1				1	1					1	1	1	
P <sub>61</sub>			0.9				0.1	0.8			0.3			0.6
P <sub>67</sub>	1	1	0.1		1	1	0.9	0.2	1		0.7	1	1	0.4
P <sub>68</sub>				1						1				
P <sub>71</sub>	0.8	0.6	0.7		0.3		0.8		0.8		0.9	0.2		0.9
P <sub>72</sub>	0.1	0.2		0.3	0.5	0.2		0.7	0.1	0.6		0.5	0.2	
P <sub>78</sub>				0.7		0.8				0.4			0.8	
P <sub>81</sub>				0.9		0.7				0.8			0.6	

Таблиця 2

варіант	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	100	200	120	320	250	300	100	100	200	150	100	100	300	100
2	200	150	120	200	300	200	400	200	300	150	200	120	200	200
3	120	300	150	200	200	300	500	100	200	200	100	200	300	300
4	300	250	200	150	150	800	400	200	100	200	250	300	200	300
5	100	200	300	150	300	300	300	300	250	400	500	300	250	250
6	300	300	600	300	400	200	250	300	250	300	300	150	200	150
7	100	800	300	100	100	200	200	400	300	200	200	100	150	200
8	-	-	-	800	-	900	-	-	-	200	-	-	200	-

варіант	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	200	400	240	640	500	600	200	200	400	300	200	200	600	200
2	400	300	240	400	600	400	800	400	600	300	400	240	400	400
3	240	60	300	400	400	600	900	200	400	400	200	400	600	600
4	600	500	400	300	300	900	800	400	200	400	500	600	400	600
5	200	400	600	300	600	600	600	600	500	800	900	600	500	500
6	600	600	900	600	400	400	500	600	500	600	600	300	400	300
7	200	700	600	200	100	400	400	800	300	400	400	100	150	400
8	-	-	-	900	-	800	-	-	-	700	-	-	600	-

Затемнення комірки означає, що відповідний оператор являється оператором вводу/виводу