



Упражнения: Алчни алгоритми

Задача 1. Задача за възлагане на дейности

Даден е списък с дейности, както и тяхното начално и крайно време за изпълнение. Да се намери максималния брой неконфликтни дейности, които човек или машина може да извърши, при условие, че участващият човек или машина може да извършва само една дейност в даден момент. За всеки две дейности се казва, че са неконфликтни, ако началното време на едната дейност е по-голямо или равно на крайното време на другата дейност.

Пример:

Вход			Изход		
Дейност	Начало	Край	Дейност	Начало	Край
a1	1	3	a3	1	2
a2	0	4	a7	3	5
a3	1	2	a6	5	8
a4	4	6			
a5	2	9			
a6	5	8			
a7	3	5			
a8	4	5			

Подсказки:

1. Подредете дейностите по време на завършване във възходящ ред. Ще използваме примерния вход даден по-горе.

Дейност	a1	a2	a3	a4	a5	a6	a7	a8
Начало	1	0	1	4	2	5	3	4
Край	3	4	2	6	9	8	5	5

2. Изберете първата дейност

Сортирани дейности	a3	a1	a2	a7	a8	a4	a6	a5
Начало	1	1	0	3	4	4	5	2
Край	2	3	4	5	5	6	8	9

3. Изберете следващата дейност, чието начално време е по-голямо или равно на времето за завършване на първоначално избраната дейност. В нашия



случай края на дейност а3 съвпада с началото на дейност а5. Намерен е един възможен отговор- а3, а5.

Сортирани дейности	a3	a1	a2	a7	a8	a4	a6	a5
Начало	1	1	0	3	4	4	5	2
Край	2	3	4	5	5	6	8	9

4. Повторете стъпка 3, докато всички дейности са проверени. Намираме следващата дейност, чието начално време е равно или по-голямо от 2. Това е дейност а7. Дейност а7, завършва в 5. Търсим следващата дейност, чието начално време е по-голямо или равно на 5. Това е дейност а6. Тя завършва в 8. Нямаме следваща дейност, чието начално време е по-голямо или равно на 8.

Сортирани дейности	a3	a1	a2	a7	a8	a4	a6	a5
Начало	1	1	0	3	4	4	5	2
Край	2	3	4	5	5	6	8	9

Отново проверяваме стъпка 3. С по-големи стойности като начало на следващи дейности след а3 са а8 и а4, но техният брой е 2. И затова отговора на задачата с конкретните входни данни е а3, а7 и а6 (максималният брой дейности, които са неконфликтни са 3).

Задача 2. Задача за магнитната лента

Дадени са n програми с дължини l_1, l_2, \dots, l_n и магнитна лента с последователен достъп. Последователният достъп означава, че за да се прочете запис, намиращ се на дадена позиция върху лентата, трябва да се премине (да се превърти лентата) до указаното място.

K_1	K_2	K_3	K_4	...	K_n
l_1	l_3	l_2	l_{k4}	...	l_{kn}

Програма K_1 е с дължина 1, K_2 - с дължина 3, K_3 - с дължина 2. За да бъде прочетена третата програма (с дължина l_2), трябва да се "превъртят" всички програми преди нея — тези с дължина l_1 и l_3 .

Дадени са вероятностите на изпълнение на всяка програма $p_1, p_2, \dots, p_n, 0 \leq p_i \leq 1$,

$1 \leq i \leq n, \sum_{i=1}^n p_i = 1$. Например, вероятност 0.5 означава, че програмата се



изпълнява толкова често, колкото всички останали програми, взети заедно. При фиксирано подреждане K_1, K_2, \dots, K_N на програмите върху лентата средното време T , необходимо за зареждането на произволна програма, се дефинира по следния начин:

$$T = \frac{\sum_{i=0}^n \left(p_{ki} \sum_{j=1}^i l_{kj} \right)}{n}$$

Задачата е да се намери подреждане на програмите върху лентата, минимизиращо средното време.

Пример:

Вход				Изход
	програма	дължина	вероятност на изпълнение	1 3 4 2
1	1	0.3		
2	3	0.1		
3	2	0.5		
4	4	0.2		

Подсказки:

Използвайте факта, че колкото по-често се използва една програма, толкова по-напред върху лентата трябва да бъде записана тя. В примерния вход по-горе програмите ще са в следния ред: 3, 1, 4, 2.

Другият определящ критерий е дължината на програмата — колкото по-дълга е тя, толкова по-назад трябва да бъде върху лентата (за да "пречи" колкото се може по-малко, когато се наложи да се превърта), т.е. подредбата според този критерий в конкретния пример ще е: 1, 3, 2, 4.

Има нареждане на програми, където не е изпълнено, че $p_{ki}/l_{ki} \geq p_{ki+1}/l_{ki+1}$. Това означава, че $p_{ki} \cdot l_{ki+1} < p_{ki+1} \cdot l_{ki}$. Отговорът значително ще се подобри, ако се разменят местата на програмата i и $i+1$, т.е. ако прилагаме тази операция ще стигнем до сортиран масив по p/l , който е най-оптималния. Записваме последователно програмите върху лентата, като на всяка стъпка избираме програмата K_i , за която отношението p_{ki}/l_{ki} е максимално.

Така получаваме следните окончателни данни. За програма 1 - $0.3/1=0.3$. За програма 2 - $0.1/3 = 0.03$, за програма 3 - $0.5/2 = 0.25$ и за програма 4 - $0.2/4=0.05$. Подреждаме резултатите (съответстващите им програми) в низходящ ред.



Задача 3. Задача за подреждане на работа в срок

Разполагате с n работни места (j_1, j_2, \dots, j_n). За всяко работно място са дадени краен срок на изпълнение (d_1, d_2, \dots, d_n) и каква печалба носи (p_1, p_2, \dots, p_n). Задачата е да се спечели максимална печалба, като в даден момент може да се назначи или обработва само една работа. Печалба има само когато работата е завършена на или преди крайния срок. Предполага се, че всяка работа отнема единица време за завършване.

Пример:

Вход			Изход
5			$j_2 \ j_1 \ j_3$
1	2	60	Максимална печалба 180
2	1	100	
3	3	20	
4	2	40	
5	1	20	

Подсказки:

1. Нека имаме n работни места j_1, j_2, \dots, j_n , които са свързани с краен срок d_1, d_2, \dots, d_n и печалба p_1, p_2, \dots, p_n .

индекс	1	2	3	4	5
работно място	j_1	j_2	j_3	j_4	j_5
краен срок	2	1	3	2	1
печалба	60	100	20	40	20

2. Сортирайте работните места според печалбата си в низходящ ред. Ако две или повече работни места имат една и съща печалба, ги сортирайте според вписването им в списъка за работа.

индекс	1	2	3	4	5
работно място	j_2	j_1	j_4	j_3	j_5



краен срок	1	2	2	3	1
печалба	100	60	40	20	20

3. Изберете първата работа. Тя има краен срок 1. Това означава, че трябва да завършим работа j2 във време 1, ако искаме да вземем нейната печалба. Вземете следващата работа, чиито краен срок е по-голям от 1. На всяка стъпка можете да изпълнявате само една задача. Взимаме j1. Тя има краен срок 2. Това означава, че трябва да завършим работа j1 на или преди време 2, за да спечелим печалбата.

индекс	1	2	3	4	5
работно място	j2	j1	j4	j3	j5
краен срок	1	2	2	3	1
печалба	100	60	40	20	20

4. Следващата по ред работа е j4, но не можете да я изберете, тъй като нейния краен срок е равен на крайния срок на вече избраната с по-висока печалба работа j1. Взимаме работа j3, тя има краен срок 3. Това означава, че трябва да завършим работа j3 на или преди време 3, за да спечелим печалбата.

индекс	1	2	3	4	5
работно място	j2	j1	j4	j3	j5
краен срок	1	2	2	3	1
печалба	100	60	40	20	20

Необходимо е да се проверява, дали задачите за изпълнение до i-тата стъпка (за всяко $i = 1, 2, \dots, n$) не стават повече от i (на всяка стъпка може да се изпълнява по една задача).

Така стигаме до максимална печалба 180 и наемане на три работни места.

Задача 4. Задача за множествата

Нека са дадени две множества. Първото е така наречената Вселена, а второто е множество от подмножества. Множеството от подмножества съдържа всички елементи от Вселената (няма други елементи), като някои могат и да се повтарят. Задачата е да се намери най-малкото множество от подмножества, които да съдържат всички елементи от Вселената.



Пример:

Вход	Изход
Вселена: 1, 2, 3, 4, 5 Брой на подмножествата: 4 1 2, 4 5 3	(4) подмножества: { 2, 4 } { 1 } { 5 } { 3 }
Вселена: 1, 2, 3, 4, 5 Брой на подмножествата: 4 1, 2, 3, 4, 5 2, 3, 4, 5 5 3	(1) подмножество: { 1, 2, 3, 4, 5 }
Вселена: 1, 3, 5, 7, 9, 11, 20, 30, 40 Брой на подмножествата: 6 20 1, 5, 20, 30 3, 7, 20, 30, 40 9, 30 11, 20, 30, 40 3, 7, 40	(4) подмножества: { 3, 7, 20, 30, 40 } { 1, 5, 20, 30 } { 9, 30 } { 11, 20, 30, 40 }

Подсказки:

На всяка стъпка взимайте подмножеството, което съдържа най-много елементи от Вселената и което съдържа елементи, които все още не сте взели. На първата стъпка винаги взимайте подмножеството с най-голям брой елементи. Когато вземете подмножеството, премахнете всички елементи в него от Вселената.

Нека разгледаме третия примерен вход:

Вселена: 1, 3, 5, 7, 9, 11, 20, 30, 40
Брой на подмножествата: 6
20
1, 5, 20, 30
3, 7, 20, 30, 40
9, 30
11, 20, 30, 40
3, 7, 40

1. Сортирайте подмножествата в низходящ ред в зависимост от броя на елементите, които съдържат

3, 7, 20, 30, 40
1, 5, 20, 30
11, 20, 30, 40
3, 7, 40
9, 30
20



2. Вземете подмножеството с най-голям брой елементи (т.е. първото подмножество). Премахнете всички елементи в него от Вселената.

Вселена: 1, 5, 9, 11

Подмножества:

1, 5, 20, 30

11, 20, 30, 40

3, 7, 40

9, 30

20

3. Вземете следващото подмножество с най-голям брой елементи. Премахнете всички елементи в него от Вселената.

Вселена: 9, 11

Подмножества:

11, 20, 30, 40

3, 7, 40

9, 30

20

4. Вземете следващото подмножество с най-голям брой елементи. Премахнете всички елементи в него от Вселената.

Вселена: 9

Подмножества:

3, 7, 40 (тук не съществуват елементи, които все още не сме взели)

9, 30

20

5. Вземете следващото подмножество с най-голям брой елементи. Премахнете всички елементи в него от Вселената.

Вселена: 9

Подмножества:

9, 30

20

6. Вземете следващото подмножество с най-голям брой елементи. Премахнете всички елементи в него от Вселената.

Вселена: \emptyset

Подмножества:

20

7. Алгоритъмът спира, тъй като Вселената е празно множество.