

**ИЗПИТ ПО ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ  
ЗА СПЕЦИАЛНОСТ “КОМПЮТЪРНИ НАУКИ” — 2. КУРС, 1. ПОТОК  
(СУ, ФМИ, 26 ЮЛИ 2020 Г.)**

**Указания:**

- 1) Имената на всички алгоритми и структури от данни да са на български език.
- 2) Ако сортирате някакви данни, уточнявайте вида на използваната сортировка.
- 3) Решенията да бъдат описани подробно.
- 4) Можете да използвате наготово всички алгоритми, изучени на лекции.

**Задача 1.** Съставете алгоритъм със сложност по време  $O(n)$  и по памет  $O(1)$ , който пресмята броя на всички сборове с цяла стойност  $n \geq 0$ , съставени само от единици, двойки и тройки. Редът на събираемите има значение. Допускат се сборове с едно събираемо и без събираеми. Например числото  $n = 3$  притежава четири представяния:  $1 + 1 + 1$ ,  $2 + 1$ ,  $1 + 2$  и самото  $3$ .

Опишете алгоритъма на Си или на псевдокод като функция `int F(int n)`.

**Задача 2.** Масивите  $P[1 \dots n]$  и  $Q[1 \dots n]$  се състоят от положителни реални числа (цели или дробни).  $P[1 \dots n]$  съдържа размерите на  $n$  парчета кашкавал;  $Q[1 \dots n]$  — желанията на  $n$  клиенти:  $k$ -тият клиент настоява да получи парче с размер, не по-малък от  $Q[k]$ . Предложете алгоритъм, който за време  $O(n \log n)$  раздава парчетата кашкавал по такъв начин, че всички клиенти да са доволни, или установява, че това е невъзможно. Опишете алгоритъма на псевдокод.

**Задача 3.** Търговска компания е произвела нова марка стока и е проучила настроенятията сред група хора. Търговската компания разполага със сведения от вида: “Ако  $A$  купи от новата стока, то след  $d$  дена  $B$  също ще купи от нея.” Един купувач може да бъде повлиян и да повлияе на няколко души от групата. Числата  $d$  зависят от  $A$  и  $B$  и са неотрицателни реални числа (вкл. дробни). Един от рекламните агенти на търговската компания съобщава, че клиентът  $X$  от въпросната група току-що си е купил от новата стока. Търговската компания иска да разбере за всеки от групата след колко дни най-рано ще купи от стоката.

Представете входните данни като граф  $G$ ; обяснете какво представляват върховете и ребрата, посоките и теглата на ребрата (ако имат посоки и тегла). Предложете най-бърз алгоритъм, който по даден граф  $G$  и първи купувач  $X$  намира за всеки човек от проучената група след колко дена най-рано ще купи от новата стока. Изберете с обосновка някой известен алгоритъм върху графи.

**Задача 4.** Докажете, че алгоритмичната задача Problem4 е **NP**-пълна.

Вход: 1) масив  $A[1 \dots n]$  от цели положителни числа;

2) масив  $B[1 \dots m]$  от цели неотрицателни числа.

Въпрос: Съществуват ли подмножество на  $A$  и непразно подмножество на  $B$  с равни сборове?

Всички алгоритми (вкл. редукцията) да се опишат на псевдокод или на Си.

## РЕШЕНИЯ

**Задача 1** се решава чрез *динамично програмиране*: първо пресмятаме броя на начините за представяне на целите неотрицателни числа, по-малки от  $n$ . Поради ограничението за паметта няма да пазим всички пресметнати бройки, а само последните три.

Описание на алгоритъма на псевдокод:

$F(n)$ : цяло неотрицателно число)

- 1)  $a \leftarrow 0$  // Брой представяния на числото  $-2$ .
- 2)  $b \leftarrow 0$  // Брой представяния на числото  $-1$ .
- 3)  $c \leftarrow 1$  // Брой представяния на числото  $0$ .
- 4) **for**  $k \leftarrow 1$  **to**  $n$  **do**
- 5)      $d \leftarrow a + b + c$  // Брой представяния на числото  $k$ .
- 6)      $a \leftarrow b$  // Брой представяния на числото  $k-2$ .
- 7)      $b \leftarrow c$  // Брой представяния на числото  $k-1$ .
- 8)      $c \leftarrow d$  // Брой представяния на числото  $k$ .
- 9) **return**  $c$  // Брой представяния на числото  $n$ .

Алгоритъмът има сложност по време  $\Theta(n)$  заради цикъла с начало на ред № 4. Сложността по памет е  $\Theta(1)$ , тъй като локалните променливи  $a$ ,  $b$ ,  $c$  и  $d$  са от примитивен тип (стойностите им са цели числа).

**Задача 2** може да се реши с помощта на *сортиране*. Понеже се изисква времева сложност  $O(n \log n)$ , трябва да използваме някоя бърза сортировка, например пирамидално сортиране.

Описание на алгоритъма на псевдокод:

Distribute( $P[1\dots n]$ ,  $Q[1\dots n]$ )

- 1) Sort( $P[1\dots n]$ )
- 2) Sort( $Q[1\dots n]$ )
- 3) **for**  $k \leftarrow 1$  **to**  $n$  **do**
- 4)     **if**  $P[k] < Q[k]$
- 5)         **return** false // Няма как всички да са доволни.
- 6)     **print**  $P[k]$ ,  $Q[k]$  // Даваме парче №  $k$  на клиент №  $k$ .
- 7) **return** true // Всички клиенти са доволни.

**Задача 3.** Групата от хора може да бъде представена чрез ориентиран граф: върхове са хората, ребра са влиянията между тях. По-точно, всяко сведение от вида: “Ако  $A$  купи от новата стока, то след  $d$  дена  $B$  също ще купи от нея.” се представя като ребро с тегло  $d$  от върха  $A$  към върха  $B$ .

Търсим най-къси пътища от дадения връх  $X$  до всички върхове на графа: дължината на най-къс път от  $X$  до  $Y$  показва след колко дена  $Y$  ще си купи новата стока. Щом теглата на ребрата са неотрицателни, можем да използваме **алгоритъма на Дейкстра**, но не и обхождане в ширина (графът е тегловен).

**Задача 4.** Това, че алгоритмичната задача Problem4 е NP-трудна, се доказва чрез полиномиална редукция  $\text{SubsetSum} \leq_P \text{Problem4}$ :

Описание на редукцията на псевдокод:

$\text{SubsetSum}(A[1 \dots n], S)$

- 1)  $m \leftarrow 1$
- 2)  $B[1 \dots m]$ : масив с един елемент
- 3)  $B[1] \leftarrow S$
- 4) **return** Problem4( $A[1 \dots n], B[1 \dots m]$ )

Редукцията се състои в тълкуването на числото  $S$  като масив с един елемент. Тази редукция е полиномиална, защото първите три реда се изпълняват за константно време (а то е полином на  $n$  от нулева степен).

Коректност на редукцията:

Функцията  $\text{SubsetSum}(A[1 \dots n], S)$  връща “истина”.

$\Leftrightarrow$  (От ред № 4.)

Problem4( $A[1 \dots n], B[1 \dots m]$ ) връща “истина”.

$\Leftrightarrow$  (От постановката на задачата Problem4.)

Има подмножество на  $A[1 \dots n]$  и непразно подмножество на  $B[1 \dots m]$  с равни сборове.

$\Leftrightarrow$  (От ред № 1 и ред № 2 следва, че  $B[1 \dots m] = (S)$ .)

Има подмножество на  $A[1 \dots n]$  и непразно подмножество на  $\{S\}$  с равни сборове.

$\Leftrightarrow$  ( $\{S\}$  е единственото непразно подмножество на  $\{S\}$ .)

Има подмножество на  $A[1 \dots n]$  със сбор  $S$ .

Доказахме, че функцията  $\text{SubsetSum}(A[1 \dots n], S)$  връща “истина”  $\Leftrightarrow$  съществува подмножество на  $A[1 \dots n]$  със сбор  $S$ .

Тази еквивалентност точно съвпада с постановката на задачата SubsetSum, следователно редукцията е коректна.

За да докажем, че  $\text{Problem4} \in \mathbf{NP}$ , трябва да съставим бърз алгоритъм за проверка на предложено решение. Сертификат ще бъдат двете подмножества на масивите  $A[1 \dots n]$  и  $B[1 \dots m]$ , за чието съществуване се пита в  $\text{Problem4}$ . Ще представим подмножествата чрез логически масиви  $P[1 \dots n]$  и  $Q[1 \dots m]$ .

```
CheckProblem4 (A[1...n], B[1...m], P[1...n], Q[1...m])
1) sumA ← 0
2) sumB ← 0
3) nonemptyB ← false
4) for k ← 1 to n do
5)     if P[k]
6)         sumA ← sumA + A[k]
7) for k ← 1 to m do
8)     if Q[k]
9)         sumB ← sumB + B[k]
10)     nonemptyB ← true
11) return nonemptyB and (sumA = sumB)
```

Сертификатът е къс, тъй като общата дължина  $n + m$  на масивите  $P$  и  $Q$  е полином (от първа степен) на дължината на входа на задачата  $\text{Problem4}$  — масивите  $A$  и  $B$ . Алгоритъмът за проверка на предложено решение се изпълнява за време  $\Theta(n + m)$  при всякакви входни данни, тоест времето на алгоритъма е полином (от първа степен) на дължината на входа на задачата  $\text{Problem4}$ .

Щом има къс сертификат и бърз алгоритъм за проверка, то  $\text{Problem4} \in \mathbf{NP}$ . Тъй като  $\text{Problem4}$  е  $\mathbf{NP}$ -трудна задача от  $\mathbf{NP}$ , то тя е  $\mathbf{NP}$ -пълна.

## СХЕМА ЗА ТОЧКУВАНЕ

Всяка задача носи по 20 точки с изключение на задача 4. Тя носи 40 точки:  
— 20 точки за установяване, че  $\text{Problem4} \in \mathbf{NP}$ ;  
— 20 точки за доказателство, че  $\text{Problem4}$  е  $\mathbf{NP}$ -трудна задача.