ДОЛНИ ГРАНИЦИ ЗА СЛОЖНОСТ. NP-ПЪЛНИ ЗАДАЧИ

ПРИМЕРНО КОНТРОЛНО № 5 ПО “ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ” — СУ, ФМИ

(ЗА СПЕЦИАЛНОСТ “КОМПЮТЪРНИ НАУКИ”, 1. ПОТОК, ФЕВРУАРИ – ЮНИ 2018 Г.)

Задача 1. Разглеждаме алгоритмичната задача за разпознаване `MaxSat`:
Вход: 1) конюнктивна нормална форма `F` със `n` клаузи
и 2) цяло число `k` от 1 до `n`.
Въпрос: Съществува ли присвояване на логически стойности на променливите, което удовлетворява поне `k` от клаузите на `F` ?

Докажете, че задачата `MaxSat` е NP-пълна. За тази цел:

а) Докажете с подходяща редукция, че задачата `MaxSat` е NP-трудна.

б) Докажете с подходящ алгоритъм за проверка, че задачата `MaxSat` $\in$ NP.

Задача 2. Разглеждаме алгоритмичната задача `Матрьошки`:
Дадени са три масива от положителни числа $L[1\ldots n]$, $W[1\ldots n]$, $H[1\ldots n]$, състоящи се от дължините, ширините и височините на n играчки с форма на правоъгълен паралелепипед, които могат да се вложат една в друга.
Търси се редът на влагане: първа да е най-вътрешната играчка.

Докажете, че задачата `Матрьошки` има времева сложност $\Theta(n \log n)$ в най-лошия случай. За целта:

а) Докажете горната граница $O(n \log n)$, като съставите бърз алгоритъм.

б) Докажете долната граница $\Omega(n \log n)$ чрез подходяща редукция.

РЕШЕНИЯ

Задача 1. Построяваме полиномиална редукция `Sat` $\propto$ `MaxSat`:

```
Sat(F: конюнктивна нормална форма с n клаузи)
1) k ← n
2) return MaxSat(F,k)
```

(15 точки)

Редукцията е коректна, защото
алгоритъмът `Sat(F)` връща истина

$\Updownarrow$ (от ред № 2)

`MaxSat(F,k)` връща истина

$\Updownarrow$ (от ред № 1)

`MaxSat(F,n)` връща истина

$\Updownarrow$ (от дефиницията на `MaxSat`)

съществува присвояване на логически стойности на променливите, което удовлетворява поне `n` от клаузите на `F`, т.е. всички клаузи на `F`.

И така, доказахме, че алгоритъмът `Sat(F)` връща истина $\Leftrightarrow$ съществува присвояване, което удовлетворява всички клаузи на `F`.

Това съвпада с дефиницията на задачата `Sat`, ето защо алгоритъмът `Sat` работи правилно, тоест редукцията е коректна. **(5 точки)**

Редукцията е полиномиална, защото присвояването `k ← n` се извършва за константно, следователно полиномиално време. Дори ако приемем, че броят `n` на клаузите не е даден, можем да го намерим с едно преминаване по клаузите на конюнктивната нормална форма `F`. При такова тълкуване редукцията има линейна сложност, но пак е полиномиална. **(5 точки)**

От наличието на полиномиалната редукция `Sat` $\propto$ `MaxSat` и от това, че задачата `Sat` е NP-пълна, следва, че задачата `MaxSat` е NP-трудна.

```
CheckMaxSat(
  F: конюнктивна нормална форма с n клаузи;
  k: цяло число от 1 до n включително;
  Certificate[1...m]: логически масив)
// m = броя на различните променливи на F.
// Certificate[j] е стойността на j-тата променлива.
// Масивът Certificate е наборът от стойности, за който
// се предполага, че удовлетворява поне k клаузи на F.
```

(10 точки)

```
Стъпки на алгоритъма:
Прочитаме F, като заместваме срещнатите променливи
с техните стойности от масива Certificate
и извършваме необходимите логически операции.
За всяка удовлетворена клауза намаляваме k с единица.
Ако k стане 0, проверката връща “истина”:
удовлетворени са достатъчно клаузи на F.
Ако четенето на F завърши с положително k,
тогава проверката връща “неистина”:
не са удовлетворени достатъчно клаузи на F.
```

(10 точки)

Този алгоритъм извършва проверката с едно обхождане на `F`, тоест има линейна, следователно полиномиална времева сложност. А щом предложено решение на задачата `MaxSat` може да бъде проверено за полиномиално време, то тази задача принадлежи на класа NP. **(5 точки)**

След като задачата `MaxSat` е NP-тррудна и принадлежи на NP, следва, че тя е NP-пълна.

Задача 2. Алгоритмичната задача `Матрьошки` се решава за време $O(n \log n)$, като сортираме играчките по обем, например с пирамидално сортиране. **(15 точки)**

Пресмятането на обемите изисква време n, а сортирането — време $n \log n$. Така цялото необходимо време е $O(n \log n)$. **(10 точки)**

Долната граница $\Omega(n \log n)$ се доказва с редукция от задачата за сортиране на положителни числа. Както знаем, тази задача изисква време $\Omega(n \log n)$. Самата редукция изглежда така:

```
SortPositive(A[1...n]: array of positive numbers)
L[1...n], W[1...n], H[1...n]: arrays of positive numbers
for k ← 1 to n do
   L[k] ← A[k]
   W[k] ← A[k]
   H[k] ← A[k]
return Матрьошки(L[1...n], W[1...n], H[1...n])
```

(15 точки)

За удобство на записа приемаме, че двете функции връщат пермутацията, която сортира елементите на съответния масив.

Редукцията е коректна, защото в случая матрьошките представляват кубове, а те се влагат в нарастващ ред на дължините на ръбовете си. **(5 точки)**

Редукцията (която се състои в попълването със стойности на трите масива с дължините, ширините и височините на играчките) се извършва за време n, което е строго по-малко по порядък от $n \log n$. Следователно редукцията е достатъчно бърза за целите на доказателството. **(5 точки)**

ТОЧКУВАНЕ

Всяко подусловие носи 25 точки, разпределени, както е указано в решенията. Цялото контролно съдържа 100 точки.