

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА СЪСТЕЗАНИЕ

Структурата от данни, в която съхраняваме точките и трасето на състезанието, е следната:

```
#define MAXARROWS 100    // Максимален брой стрелки
#define MAXPOINTS 50     // Максимален брой точки
int PtsNum, ArrNum;      // Брой точки, стрелки за тек. пример

int Arrows[MAXARROWS][2]; // Двумерен масив за съхраняване на
                          // стрелките
i-тата стрелка (0 <= i <= ArrNum -1) е с начало точка с номер
Arrows[i,0] и край - точка с номер Arrows[i,1].
```

Следните два масива се използват при решаването на задачата:

```
bool Unavoid[MAXPOINTS];
bool Reachable[MAXPOINTS];
```

Unavoid[N] приема стойност **true**, ако точка с номер N е неизбежна точка и **false** в противен случай.

Функция ReadInput() въвежда входните данни и формира масива Arrows. Определя броя на точките (PtsNum) и броя на стрелките (ArrNum).

Функция compute_results() намира неизбежните точки.

Функция PrnOutput() отпечатва резултатите.

Функция compute_results() проверява последователно всички точки с изключение на входната и изходната.

За да бъде определено дали текущата точка е неизбежна, се определя множеството S на всички точки, които могат да се достигнат от началната точка, по пътища, които не включват текущата точка. Ако крайната точка не принадлежи на множеството S, то текущата точка е неизбежна. Множеството S се определя, посредством следното обръщение към функция:

find_Reachable(CurrPnt); където CurrPnt е текущата точка.

Резултатът се запазва в масива bool Reachable[.].

След извикването на find_Reachable(CurrPnt), стойността на

Масива Reachable[K] е **true**, ако точка K може да бъде достигната от стартовата точка по път, който не минава през текущата точка. По премълчаване масивът Reachable[CurrPnt] е false. След връщането на управлението във функция compute_results() се попълва масива Unavoid[CurrPnt] със стойност **true**, ако точката е неизбежна, и **false** в противен случай.

Автор: Пано Панов