**АНАЛИЗ на задача РОБОТИ**

Първо да научим роботите да намират краищата на средната третина от низа.

Единственият начин, по който робот може да разграничи някоя междинна позиция в низа, която е далеч от краищата му, е срещата с другия робот там.

На този етап самите символи в низа нямат значение, важен е техният брой. Да разгледаме тогава низа като отсечка *PQ*. Нека *R*1 и *R*2 са двете точки, които разделят тази отсечка на три равни части. Да изпратим първия робот от *P* към *R*2, а втория от *P* към *Q*, след което от *Q* обратно към *R*2. Ако вторият робот се движи два пъти по-бързо от първия, те ще пристигнат в *R*2 в един и същи момент и срещата им ще означава, че са на правилното място.

Ето един набор от инструкции, който реализира този план (това е *Фаза 1* в решението):

Robot 1

L??0000->S0001

L??0001->R0000

IO?0000->S0001

IO?0001->R0000

IT?0000->N

IT?0001->S0010

R??0000->N

Първият робот се движи надясно, като пропуска по един ход на всеки от символите, докато срещне другия робот. Ако той види втория робот на неправилен ход (по време на „престой“ върху буквата) или изобщо не срещне втория робот, тогава броят на буквите в низа не се дели на 3 и този робот обявява низа за „грозен“.

Robot 2

L??0000->R0000

I??0000->R0000

R??0000->L0001

IO?0001->L0001

IT?0001->R1111

L??0001->N

Вторият робот се движи надясно „с пълна скорост“, докато стигне десния край, след което тръгва обратно също толкова бързо, с цел да срещне първия робот.

Нека низът има дължина 3*k* и символите в него са индексирани от 0 до 3*k*−1. Тогава след края на *Фаза 1* първият робот ще е на позиция 2*k*−1 с памет, установена на 0010, а вторият ще е на позиция 2*k* с памет, установена в 1111.

За *Фаза 2* е нужно някак да сравним броя на буквите *A* с броя на буквите *B* в средната третина на низа.

Да разгледаме по-лесна задача: как могат двата робота да сравнят броя на буквите *A* с броя на буквите *B* в целия низ?

Ключовата идея е, че роботите трябва да използват *времето* като допълнителна памет!

Ето един план, основан на тази идея. Нека единия робот да обходи низа от левия до десния край, като през всяко *A* минава направо (за един такт), а на всяко *B* изчаква един такт, т.е., движението през *B* отнема 2 такта . Вторият робот пък да прави точно обратното – да направо минава през *B* и да изчаква на *A* един такт. Тогава низът съдържа равен брой *A*-та и *B*-та тогава и само тогава, ако двата робота пристигнат в десния край в един и същи момент.

В същинската задача не можем да приложим пряко тази идея, защото не е толкова очевидно за роботите къде е краят на средната третина от низа. Затова ще модифицираме протокола от *Фаза 1* така, че времето за неговото изпълнение да дава допълнителна информация на роботите за броя на буквите *A* и *B* в средната третина на низа.

Ще казваме, че роботът се движи *по шаблон a* : *b* ако забавя времето си за *a* такта на всяко *A* и за *b* такта на всяко *B*. В описания по-горе протокол за целия низ първият робот се движи по шаблон 1 : 2, а вторият следва шаблон 2 : 1.

Да се върнем към разглеждането на низа като отсечка *PQ*. Нека двата робота току-що са се срещнали в *R*2. Да инструктираме първия да се движи от *R*2 към *P* по шаблон 1 : 2 и след това от *P* обратно към *R*1 по шаблон 2 : 1. Докато първият робот бръмчи натам и насам, нека пуснем втория от *R*2 към *Q*по шаблон 1 : 2, след което от *Q* обратно към *R*1 по шаблон 2 : 1.

Да оценим времето на всеки робот при съответното движение. Всяко *A* в средната третина на низа добавя към времето на първия робот един такт, а всяко *B* добавя два такта. При движението по първата първата третина на низа, всяко *A* и всяко *B* в нея добавят 1+2 = 2 + 1 = 3 такта.

Що се отнася до втория робот, всяко *A* в средната третина на низа добавя два такта към времето му, а всяко *B* добавя един такт. От друга страна, всяко *A* и всяко *B* при обхождането на последната третина на низа добавят към времето му 1 + 2 = 2 + 1 = 3 такта.

Следователно двата робота ще се срещнат в *R*1 тогава и само тогава, когато низът съдържа еднакъв брой букви *A* и *B* в средната си третина. Иначе те или ще се разминат, или ще се срещнат на грешен ход, или ще се срещнат някъде другаде. Ето една реализация на *Фаза 2*:

Robot 1

IOA0010->L0010

IOB0010->S0011

IOB0011->L0010

LOA0010->S0100

LOB0010->S0011

LOB0011->S0100

LOA0100->S0101

LOA0101->R0100

LOB0100->R0100

IOA0100->S0101

IOA0101->R0100

IOB0100->R0100

IT?0100->L0110

IT?0101->N

R??0100->N

Robot 2

I??1111->S0010

IOA0010->R0010

IOB0010->S0011

IOB0011->R0010

ROA0010->S0100

ROB0010->S0011

ROB0011->S0100

ROA0100->S0101

ROA0101->L0100

ROB0100->L0100

IOA0100->S0101

IOA0101->L0100

IOB0100->L0100

ITA0100->N

ITA0101->S0110

ITB0100->S0110

L??0100->N

В края на *Фаза 2* или някой робот е установил, че нещо не е наред и, следователно, е обявил низа за „грозен“, или роботите заемат две съседни позиции в низа с установена памет в 0110 за всеки от тях.

Остава им само да разберат дали са точно на позиции *k* − 1 и *k* в низа (което съответства на среща в точка *R*1 при разглеждането на низа като отсечка) или не. За тази цел те могат просто да преминат на обратно *Фаза 1*. Ако стигнат десния край на низа едновременно, кой да е от тях обявява низа за „красив“. Иначе този, който пръв стигне десния край всъщност установява, че другият се бави и обявява низа за „грозен“.

И така, *Фаза 3* на решението може да изглежда по следния начин:

Robot 1

I??0110->L0110

L??0110->R0111

I??0111->R0111

RO?0111->N

RT?0111->S0111

Robot 2

I??0110->S0111

I??0111->R0110

RO?0110->N

RT?0110->Y

\*\*\*

В заключение ще разгледаме някои алтернативни подходи към задачата и тяхното оценяване с тестовете.

Късите низове могат да се решат, като единият робот брои стъпките, а другият поддържа разликата между броя на буквите *A* и броя на буквите *B*, срещнати в средната третина на всяка стъпка. Такъв подход би преминал най-много първите три теста, където всички низове са с дължина не повече от 36.

Един низ може да е „красив“ само ако дължината му е кратна на шест. Това условие може да бъде проверено и от един робот. Ако всички такива низове се обявяват за „красиви“, а другите – за „грозни“, това ще премине само един тест, по-точно този, в който всички низове се състоят от последователни *A*-та и *B*-та.

Във *Фаза 2* първият робот може да обходи средната третина от низа от *R*2 към *R*1, като събира информация, докато вторият бърза да го срещне в *R*1. Има няколко евристики, които първият може да приложи при такъв сценарий.

Например, да следи за разликата между броя на *A*-тата и броя на *B*-тата, които е срещнал досега, и да прибегне към обявяване на низа за „грозен“, когато тази разлика заплашва да препълни паметта му. Такава стратегия ще се справи с трите теста, в които низовете са къси, с двата, в които буквите *A* и *B* следват алтернативно, както и с още два теста, специално създадени за оценка на такъв подход.

Друга идея би била да се следи разликата между броя на *A*-тата и броя на *B*-тата по модул някое малко положително цяло число *d* и да се обявява низът за „красив“ тогава и само тогава, когато броят на *A*-тата и броя на *B*-тата в средната третина на низа е сравним по модул *d*. Тази стратегия отново би се справила с трите теста, в които низовете са къси, с тези два, в които низовете са алтериращи, както и с още два, специално създадени за нея.

*Автор: Николай Белухов*