**Анализ на решението на задача**

**СТРОЕЖ**

**Първични наблюдения**

Кога един апартамент с размери може да се разположи върху апартамент с размери , ако нямаме право на разменяне на страните? Задължително трябва и , иначе ще излиза от очертанията му. Винаги има начин да се разположи, когато и двете условия са спазени. Когато се разменят страните на апартамента е аналогично – апартамент с размери може да се разположи върху апартамент с размери тогава и само тогава, когато и .

**Решения за частични точки**

**Решение за**

За тестовете, при които е полезно да се забележи, че ако има решение с разговора, то има и решение с разговора. То се постига като се направят „обратните“ разговори – разговаря се с всеки собственик на апартамент, с когото не се е разговаряло първоначално, както и наобратно. Така може да се забележи, че ни е нужен разговор, за да постигнем целта си – ако има решение с разговора при , то ще има решение с разговора. Ако има решение с и разговора при , то ще има съответно решения с и разговора. С тези наблюдения много се улеснява решението чисто имплементационно – свежда се до правене на няколко почти еквивалентни проверки.

Постигната сложност:

Имплементация: skyscraper\_22p.cpp

**Решение за**

Тук отговорът е най-много . Вече правенето на if-ове става една идея по-неприятно. Това не пречи някой от състезателите да се забавлява с това, но може да се приложи друг, не толкова времеемък подход. Вместо да се разглеждат случаите на ръка, ние може да използваме два вложени цикъла, за да разгледаме всяка двойка апартаменти, на които да „разменим“ страните.

Постигната сложност:

Имплементация с for: skyscraper\_60p.cpp

Имплементации с if: skyscraper\_40p.cpp и skyscraper\_if\_60p.cpp

**Частично решение за**

Нека дешифрираме значението на ограничението. Нека задачата има решение, в което не „разменяме“ страните на първия апартамент. Тогава може да направим „обратните“ разговори, като по този начин ще получим решение, в което „разменяме“ страните първия апартамент. Това значи, че ще има точно решение, в което разговаряме със собствениците на първия апартамент, и точно решение, в което не разговаряме. Следователно като фиксираме дали разговаряме със собствениците на първия апартамент, всички останали разговори ще бъдат точно определени. За това може да направим следното решение:

* Разглеждаме двата случая, в които разговаряме или не разговаряме със собствениците на първия апартамент.
* Тогава обхождаме всеки останал апартамент. Нека сме намерили оптимални разговори до -вия апартамент ()
  + Ако -тият апартамент не се побира върху -вия, задължително трябва да завъртим -тия, защото до -вия са наредени оптимално. Ако тогава отново не се побира, то задачата няма решение.
  + Ако -тият апартамент се побира върху -вия, то задължително няма да се побира завъртян. Иначе ще има повече от решение за текущото положение на първия апартамент. Заради това продължаваме напред.

Така може да обходим апартаментите и да преброим кои завъртаме.

Постигната сложност:

Имплементация: skyscraper\_76p.cpp

**Решение за пълния брой точки**

Нека разгледаме няколко случая касаещи апартаменти на съседни етажи.

1) Нека има апартамент с размери и апартамент с размери . Колкото и да им въртим страните, ние няма да може да разположим втория апартамент върху първия. Също може да забележим, че не може да разположим върху . С подобни разсъждения и чертаене на листче, ние може да достигнем до извода, че може да поставим апартамент с размери върху , чрез разговори, тогава и само тогава, когато и . Кратко доказателство би било: Нека разгледаме случая, когато . Тогава има два начина за поставяне на върху – да е успоредна на и да е успоредна на , или да е успоредна на и да е успоредна на . В първия случай ще е необходимо и , а във втория – и . Тъй като , винаги е по-оптимално да сложим по-голямата страна от да е успоредна на и по-малката на . Тогава трябва и . За да разгледаме случая, когато , ние може да използваме и .

2) Нека има апартамент с размери и апартамент с размери . Ние колкото и да въртим размерите, винаги ще може да разположим втория апартамент върху първия. В общия случай, ще може да разположим апартамент с размери върху , във всякакви условия, тогава и само тогава, когато . Когато липсва такава двойка съседни апартаменти, задачата има точно валидни начина за разговаряне. Когато е налична, има повече от валидни начина за разговаряне.

3) Нека има апартамент с размери и апартамент с размери . Може да се забележи, че ако се разменят страните единия, то ще трябва да се разменят страните и на другия. Това важи когато не се попада в горните два случая за апартаменти с размери и ; и .

4) Нека има апартамент с размери и апартамент с размери . Тогава ще трябва да се разменят страните на точно един от тях. В този случай се попада, винаги когато не се попада в нито един от горните три случая. При него важи и .

Остана да видим как ще довършим задачата. Нека разгледаме пример, подобен на първия тестов пример:

1)

6

5 3

3 4

2 3

1 2

1 2

2 1

2)

6

5 3

3 4

2 3

1 2

1 2

2 1

3)

6

5 3

4 3

2 3

1 2

1 2

2 1

При него веднага забелязваме, че трябва да обърнем първия или втория апартамент. След това забелязваме, че трябва да обърнем втория или третия апартамент. Четвъртият апартамент не е обвързан с третия апартамент, за това нека първо да определим какво се случва до момента. Нека оцветим първия и третия апартамент в синьо, както и да оцветим втория апартамент в червено. Има два варианта – да обърнем всички сини апартаменти или всички червени апартаменти. Тъй като броят червени е по-малък от броя сини, то е по-оптимално да обърнем всички червени апартаменти. Нека продължим с четвъртия апартамент и натам. Там забелязваме, че трябва четвъртия и петия апартамент да са обърнати еднакъв брой пъти. Също така, забелязваме, че трябва да обърнем петия или шестия апартамент. Нека оцветим четвъртия и петия апартамент в зелено, както и шестия апартамент в оранжево. Тогава имаме избор да обърнем всички зелени или всички оранжеви апартаменти. Тъй като оранжевите са по-малко на брой, то е по-оптимално да обърнем тях.

С този пример разгледахме какво би правила нашата програма. Нека да оцветяваме апартаментите в синьо и червено, като първоначално оцветим първия апартамент в синьо. После продължаваме с втория апартамент на сетне и виждаме в кой от четирите случая сме. Ако сме в 1), то ние казваме, че не може да се построи небостъргача и терминираме програмата. Ако сме в 3), то оцветяваме апартамента в същия цвят като предходния. Ако сме в 4), то ние оцветяваме в цвета, различен от този на предходния апартамент. Ако сме в 2), то ние определяме в кой цвят има по-малко оцветени апартаменти, завъртаме ги, като започваме оцветяването наново от текущия апартамент. Когато привършим и с последния апартамент, ние пак обръщаме апартаментите от по-малко срещания цвят – в примера не достигаме до 2) за да определим дали да обърнем зелените или оранжевите. С това сме решили задачата.

Постигната сложност:

Имплементация: skyscraper\_100p.cpp

*Автор: Борис Михов*