АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
COMPANY

Задачата представя напълно валиден проблем от реалния живот – наистина в една софтуерна фирма е желателно хората от един екип да стоят близо един до друг. Тук проблемът е достатъчно опростен за да съществува полиномиално решение, което го решава. В случая опростяването идва от това, че всички програмисти трябва да бъдат разположени в един единствен ред.
Нека разгледаме дървото, което представя йерархията на компанията. Можем да направим почти очевидното наблюдение, че щом директният началник на всеки програмист е наляво от него, то и индиректните му началници ще са наляво. Така стигаме до извода, че шефът ще е винаги най-отляво, което е споменато и в самото условие. Ако след него сложим всички негови директни подчинени, то някои от тях биха били отделени от екипите си, което не е желателно. Следователно ще трябва да изпечатваме екипите един по един. Тоест ако изпечатаме някой програмист, то ЗАДЪЛЖИТЕЛНО след него изпечатваме всички негови подчинени (преки или непреки) преди да изпечатаме когото и да било друг. Това трябва да подсеща състезателите за рекурсия (рекурсивно изпечатване на екипите). Всъщност това е DFS (Depth-First Search) в най-чистата му форма – изпечатване на дърво в ред корен-ляво-дясно (само че може да имаме повече от 2 деца, така че всъщност е корен-дете1-дете2-...-детеK, където K е броят деца, които има този връх).
За да не си помислят състезателите, че ги подценяваме като им даваме твърде лесни и/или стандартни задачи, в задачата са добавени няколко уловки. Първата от тях е, че трябва да изпечатаме дървото в лексикографски ред. За целта преди да пуснем рекурсията трябва да сортираме наследниците на всеки връх в нарастващ ред. Някои от тестовете (сравнително малка част от тях, която е указана в условието) съдържат много върхове, като има и „гадни" тестове, в които един връх има много наследници. За да бъдат хванати тези тестове трябва да се ползва сортиране със сложност O(N*logN) или по-добра. Но за състезателите, които не знаят вградената в езика функция за сортиране или не знаят как да пишат такива „бързи" сортировки са дадени 85 точки дори с O(N*N) такива.
Другият трик, който според мен ще е далеч по-неочевиден за повечето състезатели в тази възрастова група е, че рекурсия с дълбочина един милион има ОГРОМЕН шанс да даде stack overflow (което от системата се отчита като runtime error). Като цяло рекурсия с дълбочина повече от няколко хиляди е опасна (като rule on thumb можете да ползвате, че рекурсия над 10000 нива е опасна, което, разбира се, отново зависи от настройки на компилатора, операционна система (Linux или Windows) и колко памет заема всяко ниво на рекурсията). За целта трябва или да си увеличим стека (ако знаем как, чрез #pragma команда към компилатора), или да ползваме стек (като симулираме рекурсията итеративно), като вторият вариант е очакваният начин да се реши задачата.
Сложността на решението може да се сметне като O(N) за четене на входа и създаване на графа, O(N*logN) за сортиране на наследниците, и O(N) за самата рекурсия (или симулирането ѝ чрез стек). Общата сложност е доминирана от сортирането, тоест O(N * logN).

Автор: Александър Георгиев