АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
ОДИТ

1. ***Бавно решение***

Най-простото решение е да направите следното:
В случай на повишение на заплатата, автоматично се дава повишение и на всички подчинени.
В случай на заявка, се изчисляват минималните и максималните заплати и се извежда разликата.
Това ще ни даде линейна актуализация на времето и заявка за линейно време, която ще бъде твърде бавна за нашите нужди.

2. Оптимизации

Една малка оптимизация е да се отбележи, че повишението не е задължително да се приложи незабавно. Ако има много повишения наведнъж, можете да се кешират (запомнят). Маркира се, че са постъпили повишения, но всъщност не се извършва актуализацията, докато не пристигне заявка за търсене на коефициент на удовлетвореност.
Тогава се извършва актуализацията..
Друго е да се преизчислят минималните и максималните заплати на служителите, така че заявките могат да отнемат постоянно време вместо линейно време.

3. ***Пълно решение***

Вместо да се стремим към постоянно актуализиране или към заявка за постоянно време, ще го направим
Вместо това отидете за актуализация и заявка за логаритмично време.
Започваме с пренасочването на идентификационни номера към всички служители, като извършим препредаване дървото. Имайте предвид, че за всеки служител, всички идентификационни номера на този служител Подчинените съставляват съседен интервал.
С това, сега можем да поддържаме всички заплати в дървото, което поддържа диапазон на интервала и заявка за обхват min / max, като и двете са операции, които се изпълняват в логаритмично време.

```
struct emp{
          int sup, subcnt, id, ticket ;
} empdata[MAXN+1] ; // this is 1-indexed
int salaries[MAXN] ;
struct cache{
   int bonus, hi, lo ;
   char valid ;
} cache[C1024] ;
```

Автор: Пано Панов