АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ДАЛЕЧНИ ДЕСТИНАЦИИ

Функция lockup() чете входните данни и връща броя на летищата в int променливата na. Построява се двумерен масив dct[][], в който се записва трасовата мрежа с разстоянията между летищата. INF е константа (безкрайност) – число по-голямо от най-дългата дистанция между летища. Със стойност INF се инициализира масива dct.

Използва се обобщен алгоритъмът на Флойд, който след като приключи работа, е изчислил дължините на минималните пътища между всяка двойка върхове от графа и то без да е необходима допълнителна памет (използва се матрицата dct). Това означава, че ако в началото в dct[i][j] е записано теглото на реброто (i, j), то след изпълнение на алгоритъма на Флойд върху матрицата dct[][], стойността на dct[i][j] ще бъде дължината на минималния път между i и j.

Алгоритъмът се състои в следното: за всеки два върха i, j, на dct[i][j] се присвоява по-малкото от dct[i][j] и dct[i][k]+dct[k][j], за всеки връх на графа.

Дефинират се променливи smin=1 и smax = INF. Променливата mid = (smin + smax) / 2;
Помощната матрица work се зарежда с оператора:

work[i][j] = dct[i][j] <= mid ? 1 : INF;

Поставят се тегла 1.

Прилага се алгоритъма на Флойд за така генерираната матрица work.

Стойността на work[i][j] показва от колко ребра е съставен пътя от летище i до летище j.

Респективно, колко кацания ще бъдат извършени.

Намира се максимума на work в променливата d.

Ако:

```
if(d <= L){
    ans = mid;
    smax = mid - 1;
  } else{
    smin = mid + 1;
  }
```

Това се повтаря итеративно, smin расте, ако d>L, smax намалява, ако d<=L, докато smin стане равно на smax. Тогава в mid се получава желания резултат.

Автор: Пано Панов