**АНАЛИЗ НА ЗАДАЧА: Свързващи полети**

За всеки полет записваме най-ранния час на излитане (начален час), най-късния час на пристигане (краен час) и максималното време на изчакване (общото време, което трябва да чака Пеър в двете летища през интервала от време от началния до крайния час, за текущия полет. За целта използваме структурата:

**struct Flight {**

**int adep, ades; // departure and destination airports**

**int deptm, // the earliest departure time**

**arrtm, // the latest arrival time**

**waittime; // the maximum waiting time of flight**

};

За всяко летище поддържаме динамичен масив от състояния.

**struct State {**

**int tim; // current latest arrival time**

**int waittime; // the accumulated waiting time, till time 'tim'**

**State(int a, int b): tim(a), waittime(b) {}**

**};**

Всеки елемент на масива съдържа времето за пристигане на това летище + общото време на чакане до този момент. Списъкът е сортиран по времето на пристигане. Ако трябва да сме на дадено летище в даден момент t, можем да използваме двоично търсене, за да намерим последния момент U в списъка, който е не по-късно от t и да добавим допълнителното време на изчакване t-U към времето на изчакване в момента U. Всеки път, когато добавяме състояние към даден списък (състоянията винаги се добавят в нарастващ ред по време), ние проверяваме дали можем да получим по-малко време за изчакване, като започнем от последното състояние в списъка и изчакаме до текущото време. В някои случаи не трябва да добавяме състоянието към списъка.

Сортираме полетите в разписанието в ненамаляващ ред по време на пристигане. Ако някой полет завърши в определено време, следващият полет, който е възможно да хванем, не може да започне по-рано от този час, така че можем да разгледаме полетите в сортирания ред. За всеки полет намираме с помощта на масива от състояния на летището на излитане, времето за изчакване до началния час на полета и добавяме крайното време към масива на състоянието на летището на кацане. Ако времето на кацане е по-късно от Z, трябва да пропуснем този полет.

След като разгледаме всички полети, проверяваме дали масивът от състояния за летище P съдържа някакви елементи (не е празен). Ако е така, можем да намерим времето за изчакване до момента Z, използвайки най-новото състояние в масива. В противен случай не е възможно да се гарантира пристигане на Петър на летище X преди час Z.

Вместо да използваме масиви с динамичен размер, можем да изчислим размерите на масивите и да разпределим памет за тях в началото на програмата. Масивът на състоянията за град i не може да съдържа повече елементи от броя на полетите в разписанието, които пристигат в град i, плюс една за град 1.

Автор: Пано Панов