

Състезание 2 – Златна дивизия

Задача 1 – Анализ

Нека сортираме всички полети по най-късното време, в което пристигат в дестинацията си. За всяко летище A ще пазим списък от състояния, като всяко състояние описва, че е възможно да се стигне в даденото летище A във време T като общото чакане до момента е било W .

В самото начало списъците на всички летища са празни, защото все още не знаем в какви времена можем да стигнем в тях и с колко чакане. Единствено можем да отбележим, че в летище 1 може да се стигне във време 0 с общо чакане 0, които идва от условието.

Нека обходим полетите в реда, в който са сортирани. За всеки полет проверяваме дали не пристига по-късно от указаното време T . Ако е така, няма смисъл да го разглеждаме и няма смисъл да разглеждаме следващите полети, защото те ще имат още по-късни или равни времена на пристигане.

За всеки полет F проверяваме списъка от състояния на летище O , от което се излита. Ако този списък е празен, следователно все още не сме намерили начин да стигнем до това летище. В този случай няма как да използваме полета F . Това ни се гарантира от сортирането на полетите по най-късни време на пристигане. Щом не сме намерили полети, преди F , които да можем да свържем с него, следователно това не е възможно.

Ако списъкът има състояния проверяваме дали най-ранното време, в което сме успяли да се доберем в летището на излитане O не е по-късен от най-ранното възможно излитане на полета F . Ако е така, състоянията в този списък не ни вършат работа. Следователно този полет не може да се използва.

Нека в списъка има състояния, които отговарят на времена, които са не по-късни от най-ранното време на излитане на полета F . Трябва да се намери състоянието с най-късно време, което е не по-късно от най-ранното време на излитане на полета F . Нека това състояние има време T_s и до него се стига с общо изчакване W_s .

Ето няколко примера, които описват ситуациите от по-горе. Ако списъкът със състояния на летище A има 4 състояния с времена 40, 55, 63, 75 и полета F , който излита от A има най-ранно време на излитане 37, то явно няма как да се стигнем в летище A и след това да се хване със сигурност полет F . Ако обаче времето за полета е например 56, то има две състояния преди това време. Ние ще се интересуваме от второто, защото то е най-късното, което е не по-късно от времето на полета F . По-късно ще стане ясно защо е достатъчно да използваме само това състояние. Това ще следва от начина, по който се строи списъкът от състояния за всяко летище.

Нека сме избрали правилното състояние. Вече можем да пресметнем с колко чакане и в какво време полета F може да пристигне в дестинацията си D . Ако четирите времена от условието, описващи кога излита и каца самолета са a_i, b_i, c_i, d_i (както са зададени в условието), то времето за чакане, за да се пристигне

в летище D ще бъде $W_s + (b_i - a_i) + (d_i - c_i)$. С това време на изчакване с полет F ще се озовем в дестинацията D му във време d_i . Следователно сме намерили ново състояние за списъка на летище D . Преди да го добавим обаче е нужно да проверим дали този списък няма състояние, което има време по-ранно от d_i и което ни позволява да пристигнем с друг полет в D и просто да дочакаме да стане време d_i , като общото чакане да е по-малко. Например е възможно в D да се стига във време 300 с общо чакане от 40 минути, като с полет F се пристига в 320 с общо чакане 100 минути. Тогава очевидно е по-добре да се използва пътуването, което пристига в 300 и да се почака още 20 минути, за да стане 320. Това ще доведе до общо чакане от $40+20 = 60$ минути, което е по-добре от 100 минути, които се получават с полет F .

Понеже полетите се разглеждат в реда, в който са сортирани се гарантира, че състоянията се създават в нарастващ ред на време на пристигане. Освен това поради тази проверка, която се прави е гарантирано, че списъкът от състояния за всеки град ще е сортиран и по общо време на чакане. Това е така, защото всяко ново състояние, което се добавя има по-добро време на чакане от предишното добавено.

Накрая, последното добавено състояние за търсения град P дава решението на задачата, защото то ще е с най-кратко време на изчакване от всички състояния, които сме добавили за P .

Сложността по време на тази задача е $O(M * \log M)$. Това е така, защото за всеки полет се извършва двоично търсене сред състоянията на началното летище. Тези състояния не могат да са повече от M . Освен това сортирането на полетите може да стане със сложност $O(M * \log M)$.