**Анализ**

**Подгрупа 1**

При такива ниски ограничения е достатъчно да се направи пълно изчерпване на всички възможни пътища с обхождане в дълбочина, като за всеки път се гледат опасността и горивото. Накрая се извеждат параметрите на най-добрия път съгласно условието.

Сложност: O(N!)

**Подгрупа 2**

В тази подзадача всички скокове изискват точно 1 единица гориво и имат опасност ≤ 1000. Тъй като ограниченията са малки, можем да пробваме за всяко *p* от 1 до 1000 дали е възможно придвижването от 1 до N движейки се само между астероиди с опасност ≤ *p*. Първата стойност, при която има път до астероид N, е търсената минимална опасност. След това обаче трябва да намерим най-евтиният път от 1 до N с такава опасност. Понеже всички скокове изискват еднакво гориво, то задачата се свежда до намиране на път с най-малък брой ребра, което може да се направи с обхождане в ширина (BFS). Можем да комбинираме методите за намиране на двете стойности в едно, като за всяко *p* директно търсим най-бързият път между 1 и N. Ако такъв път няма, то продължаваме нататък със следващата стойност и така докато намерим решение.

Сложност: O((N+M)\*Pmax)

**Подзадача 3**

Тази подзадача много прилича на миналата, единствената разлика е в стойностите на *p*. Едно важно наблюдение за задачата е следното: ако при дадена опасност *p* можем да стигнем от 1 до N, то ще можем да го направим и за всяка опасност *S* ≥ *p*. Това свойство ни подсказва, че можем да използваме двоично търсене по отговора. На всяка стъпка от двоичното търсене пускаме BFS по ребрата с опасност ≤ *mid*със сложност log2(Pmax). Стойността в l и r след двоичното търсене е исканата минимална опасност, а нужното гориво е резултата от съответното BFS.

Сложност: O((N+M)\*log2Pmax)

**Подзадача 4**

Тук опасността винаги е равна на 1, но цените имат различни стойности. Това означава, че минималната опасност винаги ще бъде 1 и трябва единствено да намерим най-евтиният път до астероид N. Подходящ алгоритъм за това е алгоритъмът на Дийкстра с приоритетна опашка, който смъква сложността до O(N+M\*log2N).

Сложност: O(N+M\*log2N)

**Подзадача 5**

Решението за тази подзадача е просто комбиниране на миналите две решения. За всяка проверка за *mid* в двоичното търсене ще пускаме Дийкстра, игнорираща ребра с *p*>*mid*, и ако тя върне резултат различен от безкрайност, то съществува път от 1 до N.

Сложност: O(log2P\*(N+M\*log2N))

**Подзадача 6**

Пълното решение на задачата използва минимално покриващо дърво (MST), чрез което да намерим минималната опасност. Това работи, защото по този начин проверяваме свързаност с минимална сложност, вместо да правим пълно обхождане за всяко *p*. Добавяме ребрата от най-безопасни до най-опасни, като първото ребро, за което астероиди 1 и N са в една компонента, има търсената минимална опасност. След като знаем този минимум, отново можем да използваме една Дийкстра за откриване на минималната сума. Важно е да се отбележи, че при реализирането на MST трябва да използваме двете оптимизации за DSU – сливане по ранг/размер и компресия на пътя. Чрез тях операциите се свеждат до сложност О( α(N) ), където α е обратна функция на Акерман (функция, която расте изключително бавно; с ограниченията от задачата можем да приемем, че α(N) = 1).

Сложност: O(M\*log2M+N\*α(N)+N+M\*log2N) ≈ O(N + M\*log2M)

*Автори: Калоян Върбанов и Преслав Тошев*