Разбор задачи «Путь через горы»

Автор задачи и разбора — О. Пакуляк

Будем решать задачу динамическим программированием.

Пусть $a[i][j]$ — минимальное расстояние, которое придется пройти на пути из первой вершины в j-ю, построив не более i мостов. Тогда ответом на поставленную задачу будет $a[k][n]$. Из определения вытекает, что $1 \leq i \leq k$ и $0 \leq j \leq n$.

Обозначим через $dist(p, q)$ длину отрезка, соединяющего точки с номерами p и q.

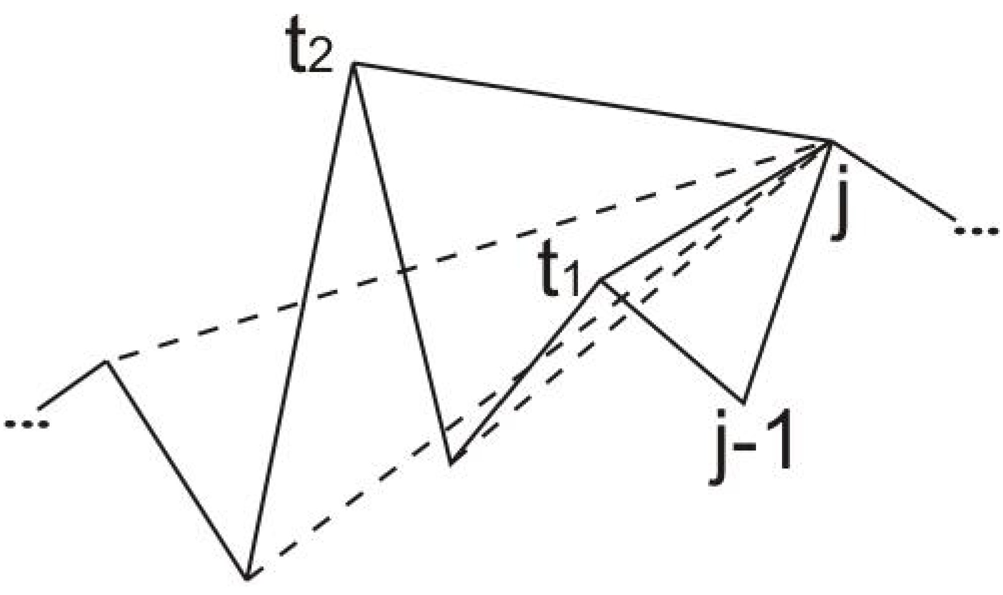
1. (Инициализация) Для корректной работы формулы, выведенной ниже необходимо задаться некоторыми начальными данными, из которых все будет вычисляться. В данном случае достаточно сказать, что $a[*][1] = 0$ ($*$ здесь означает любой индекс от 0 до k).

2. Выведем теперь для $a[i][j]$ рекуррентную формулу. Существует всего два случая попасть в вершину j: либо из предыдущей вершины по земле, либо по мосту из какой-то вершины $1 \leq t < j$ (при условии, что этому мосту не придется идти под землей и он не слишком длинный). **В первом случае** пройденное до j расстояние будет равно $a[i][j-1] + dist(j-1, j)$. **Во втором случае** для того, чтобы можно было построить мост между t и j, необходимо и достаточно выполнения двух условий: $dist(t, j) \leq R$ и «отрезок t-j не идет под землей». Допустим, что мы как-то проверили истинность обоих утверждений. Тогда $a[i][j] \leq a[i-1][t] + dist(t, j)$ ("$\leq$" — потому что возможно найдется способ и получше этого).

   Итак, рекуррентная формула такова:

   $$a[i][j] = min\Big\{a[i][j-1] + dist(j-1, j), a[i-1][t_1] + dist(t_1, j), \ldots, a[i-1][t_s] + dist(t_s, j)\Big\},$$

   где $t_1, \ldots, t_s$ — все значения t, для которых можно постоить мост из t в j.



Каждое $a[i][j]$ вычисляется за $O(j \cdot T)$, где $O(T)$ – время проверки соответствующего моста. Итого $O(Tn^2k)$.

Обсудим разные подходы к проверке моста t-j на «правильность». Непонятным местом здесь является проверка того, что он не проходит под землей, более формально — отрезок t-j лежит не ниже любой из точек $t+1$, $t+2$, $\ldots$, $j-1$. Можно для каждого t просто бежаться переменной m от $t+1$ до $j-1$ и проверять, что векторное произведение $\overrightarrow{jt}$ и $\overrightarrow{jm}$ больше либо равно нуло. В этом случае $T = j - t - 1$ и суммарное время программы будет $O(n^3 k)$. При ограничении 100 на n и k и при современных мощностях это вполне приемлимо.

Отметим, что если длина отрезка t-j превышает допустимую длину моста, то не факт, что отрезок $(t-1)$-j также будет слишком длинным. Так что это не повод завершать вычисления для данного j.

Если бы в условии задачи нельзя было бы строить мост, проходящий по поверхности земли, то при проверке его на правильность вышеописанные векторные произведения уже не могут равняться нулю, то есть все они больше 0.

Напоследок скажем пару слов о том, ускорить решение примерно в n раз. Дело в том, что на правильность отрезок можно проверять за $O(1)$. Будем перебирать t от $j-2$ до 1. При этом будем хранить и обновлять на каждом шаге q — минимальную по полярному углу точку среди $t+1$, $t+2$, $\ldots$, $j-1$ относительно j (то есть точку, которая больше всего мешает при построении моста до вершины j). Если векторное произведение $\overrightarrow{jt}$ и $\overrightarrow{jq}$ больше либо равно нулю, то отрезок t-j годится, и q надо присвоить t. В противном случае мост t-j нельзя посторить и ничего делать не нужно.