# Sky Walking (Review)

## Subtask 1

For each skywalk and each building, check if they intersect and if they do, add a new vertex for the intersection point. Additionally, put a node on the bottom of each of the buildings $s$, and $g$. Add edges between consecutive nodes on each building and also for consecutive nodes on each skywalk. Use Dijkstra to find the shortest path from the node on the bottom of building $s$ to the node at the bottom of building $g$. Number of vertices and edges are $\mathcal{O}(NM)$, so the total complexity is $\mathcal{O}(NM \log NM)$.

## Subtask 2

The solution for the previous subtask works here as well, however the graph must be built more efficiently. To do that, iterate in increasing order over heights which either contain a skywalk or the endpoint of a building. The goal is to keep a list of all buildings that are at least as tall as the current height. Given such list, for each skywalk, start from its left endpoint and move through the list. Each element is an intersection between that skywalk and a building. Hence you will at most visit 10 elements before reaching the right endpoint. Add a vertex for each intersection. The rest of the solution is the same as the last subtask. For maintaining the list of buildings, it is enough to start from a list of all buildings. Then at each height, after processing the skywalks for that height, remove the buildings that have an endpoint at that height. This can be done using a linked-list or a BST.

## Subtask 3

When $s = 0$ and $g = n - 1$, it can be proven that skywalks are always traversed from left to right. Additionally when all of the buildings have the same height, it can be shown that there exists an optimal path where each skywalk is either not visited or is traversed completely until its right endpoint (though the entrance point to that skywalk might not be its left endpoint). Iterate over skywalks in increasing order of their right endpoint, breaking ties in favor of lower skywalks. The goal is to maintain the minimum cost to reach each skywalk's right endpoint. During the iteration, for each skywalk, update this value for the skywalks just above, and just below its right endpoint based on its own value. The answer is the minimum cost to reach the right endpoint of one of skywalks ending at $n - 1$ adding the cost of reaching to the bottom of the building $n-1$.

## Subtask 4

For this subtask, the solution is to use the same strategy as the first two subtasks and build a graph. Consider the graph in those two subtasks. Let $e$ be a vertical edge connecting points $(x, y_1)$ and $(x, y_2)$ for some $x, y_1, y_2$ ($y_1 < y_2$). Edge $e$ is called irrelevant if the point $(x, y_2)$ lies strictly inside of some skywalk. Since $s = 0$ and $g = n-1$, it can be proved that there exists a shortest path that does not pass through irrelevant edges. Hence, after removing the irrelevant edges from the graph, the length of the shortest path doesn't change. In the new graph, for each vertical edge, the top node is the endpoint of a skywalk. So, there are at most $\mathcal{O}(M)$ vertical edges left in the graph. This means at most $\mathcal{O}(M)$ vertices have at least one edge connected to them. Discarding the other vertices, the same approach as the first two subtasks can be followed on the new graph.

## Subtask 5

The solution for this subtask is almost the same as the last subtask. However, for the previous theorem to hold, the skywalks must be adjusted. Specifically, for each skywalk between two buildings such as $l$, and $r$ where $l < s < r$, divide it into 3 parts as follows:

- let $a$ be the last building before $s$ (including $s$) that is as tall as this skywalk.
- let $b$ be the first building after $s$ (including $s$) that is as tall as this skywalk.
- replace the skywalk with the following skywalks:
    - skywalk connecting buildings $l$ and $a$.
    - skywalk connecting buildings $a$ and $b$.
    - skywalk connecting buildings $b$ and $r$.

The same adjustment must be done for all skywalks that intersect or pass over building $g$. Then the same solution as subtask 4 works.