ACM 2005, Problem F. Solution.

Michal Korch, Michal Szynkiewicz

January 8, 2007

1 Solution

The point of the problem is to find a minimal number of streets that we have to cross to get to University, we are not interested in real length of path. Hence we can compress map (relabel coordinates), so that the smallest coordinate is labeled 1, next – 2, and so on (pictures below).

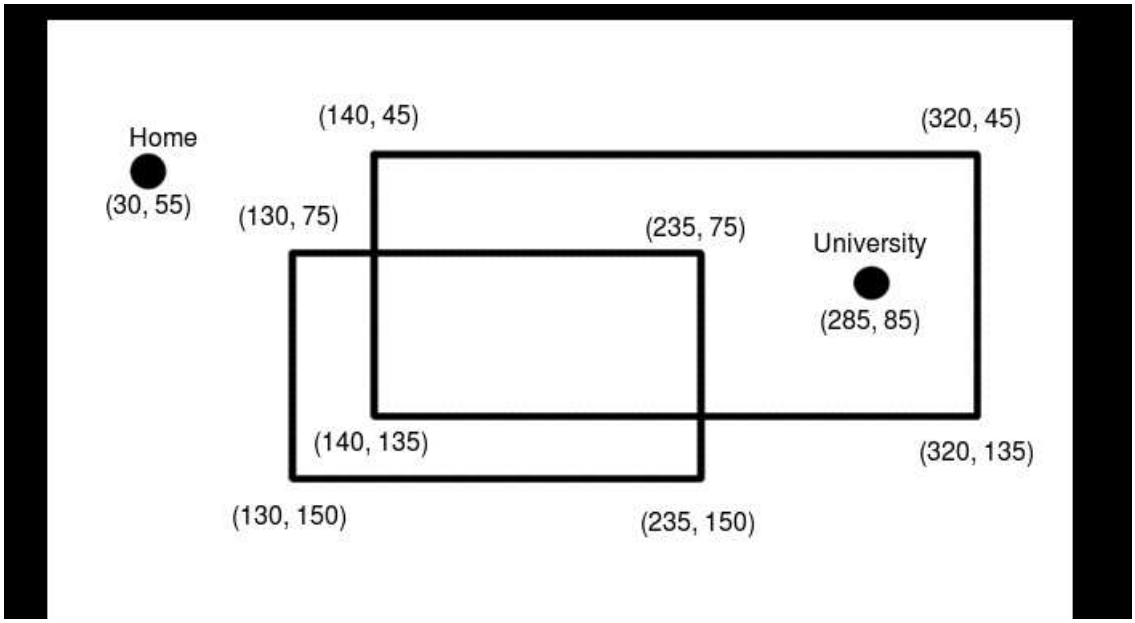

Figure 1: Before compression.

As a field (a, b) I will mean place between lines: $y = b$, $y = b - 1$, $x = a$, $x = a - 1$.
All streets are either vertical or horizontal and we can not cross streets at intersection, and the length of path is not important. Therefore we can limit moves to horizontal and vertical ones.
At this moment our problem can be solved by finding the shortest path in graph, whose vertices are all fields on map. For two fields p and q there is a edge between them if, and only if they share a side. The length of edge equals 1 when there is a street between p and q, 0 otherwise.

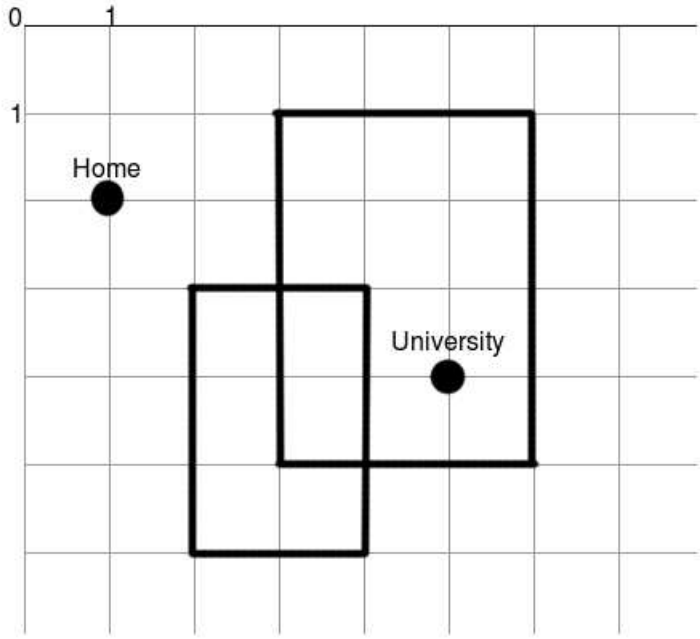

Figure 2: After compression.

To find the shortest path we will use Dijkstra algorithm. In our case all edges have length 0 or 1. Therefore the difference between priorities in queue is not bigger then 1 and we can replace usual priority queue with one that provides all necessary operations in . It can be implemented as to lists, one containing all the elements with priority equal to priority of minimal element, the other with priority bigger by 1.
Dijkstra algorithm implemented on such priority queue will take linear time (in regard of size of graph).

2 Cost analysis

Ozn n = number of coordinates.

1. Compressing (sorting and relabeling): $\Theta(nlogn)$
2. Field graph creation (in array): $\Theta(n^2)$
3. Finding shortest path: $\Theta(n^2)$

Overall cost: $\Theta(n^2)$