Figure 2: A set $S = \{\ell_1, \ldots, \ell_5\}$. Underline is their upper envelop $U(S)$. The vertices on the upper envelop are marked by tiny disks. The line $\ell_6$ is the new line being tested for calculating the new upper envelope. The algorithm checks if $e_5$ intersects $\ell_6$. As the answer is negative, it deletes $e_5$ from the envelope, and repeat the same check for $e_4$. Since the answer is negative, $e_4$ is deleted from the envelop, and the test is repeated for $e_3$. The answer is positive, and the intersection point is $q$. Then $U(S_6)$ contains the portion of $U(S_5)$ from the leftmost point till $q$, and the part of $\ell_6$ from $q$.

Hint — Define $S_i = \{s_1, \ldots, s_i\}$. Assume by induction that you have already computed the vertices of $U(S_i)$. How can you efficiently compute $U(S_{i+1})$?

Answer:
In the solution the term segment defines also a half-line, namely the portion of a line on one side of a point on this line. We say that a line $\ell$ contributes to $U(S)$ if $U(S)$ contains a segment which is on $\ell$.

It is easy to see that every line of $S$ contributes at most one segment to $U(S)$. Hence the complexity of upper envelope is $O(n)$.

To compute $U(S)$ time $O(n)$, we use the induction method. We assume that we have already computed the vertices of $U(S_i)$. We also assume that in $O(1)$ time we can check if a line intersects a segment, and find its intersection point.

Claim $\ell_{i+1}$ intersects $U(S_i)$ exactly in one point, and hence it splits $U(S_i)$ into two parts; one which is below $\ell_{i+1}$ and does not appear in $U(S_{i+1})$, and one which is above $\ell_{i+1}$ and is unchanged.