B: Deliveries

Note that for each delivery request only two destinations make sense: the ones closest to the pick-up location from the left and from the right. If there is no destination on one side, we can assume that it is $\pm\infty$. From now on, we denote the i-th delivery request by a triple (s_i, l_i, r_i): the pick-up location, the left destination and the right destination respectively.

Consider an optimal route. Let L denote the leftmost visited point, R the rightmost visited point and $F \in [L, R]$ the finish point. Suppose that the point L is visited for the first time before R. We travel from 0 to L, then from L to R and finally from R to F, possibly with some detours. In the optimal route there are no detours when travelling from R to F: once we reach the point R all packages have been picked up and they only need to be delivered.

All requests with $r_i \leq R$ are satisfied without taking detours. The same can be said about packages with $l_i \geq F$ and about packages with $s_i \leq 0$. The remaining packages have $s_i > 0$, $l_i < F$, $r_i > R$, and they need to be delivered to the left.

Let $J_{R,F} = \{[l_i, s_i] : s_i > 0 \land l_i < F \land r_i > R\}$ be a set of intervals defined for these remaining packages. Consider a maximal connected segment $[x, y]$ in the union $\bigcup J_{R,F}$. We have two cases how to handle packages in $[x, y]$:

(a) if $x > 0$ then once we reach the point y, we go back to x and then return to y;
(b) if $x \leq 0$ then we start by going from 0 to y and then we go directly to L.

In the first case, we travel additional $2(y - x)$ distance and in the second case only $2y$. This is optimal: we need to go through these segments from left to right at some point, and we don't take detours after reaching R. In total, handling these packages costs us $2|D_{R,F}|$, where $D_{R,F} = \bigcup J_{R,F} \cap [0, +\infty)$.

The final observation that we need is that for a fixed point R, the point L is determined uniquely. It is the minimum of points l_i for packages with $r_i > R$, and all pick-up locations s_i. Each of these points needs to be visited and there is no point in travelling further to the left. We denote the optimal leftmost point for given R by L_R.

We can now provide a final formula. For a fixed R and F, the total distance is:

$$2(R - L_R) - F + 2|D_{R,F}|$$

Now it's easy to solve the problem in $\mathcal{O}(n^3)$: we iterate over all possible points R and F, and compute the distance in $\mathcal{O}(n)$ time. To solve the problem in $\mathcal{O}(n^2)$, we can fix only the point R and sweep over all possible points F.

The full solution works in $\mathcal{O}(n \log n)$ time. We sweep over all possible points R from the right to the left, and keep answers for all points F in a segment tree T. More precisely, we maintain $T[F] = 2|D_{R,F}| - F$. Then we can compute the optimal total distance for a fixed R using a single range min query.

To know how to update the tree T, we additionally keep a set S of segments. Each segment represents a package that needs to be delivered from right to left. We don't keep in S segments that are contained within other segments – they don't change the answer. This allows us to keep the set S sorted by both left and right endpoints.

When we move the point R to the left, new segments may appear. Suppose a new segment $[a, b]$ appears. We first check if it is contained within some existing segment in

S. If it is, then we skip it. Otherwise, we remove from S all segments contained in $[a, b]$ and then insert segment $[a, b]$.

We still need to describe how the tree T changes. Suppose we insert a segment $P = [x, y]$ into the set S between $Q = [a, b]$ and $R = [c, d]$. This means that $a < x < c$ and $b < y < d$. Then we need to add $|P \setminus Q|$ to $T[x + 1 : c]$ and $|P \setminus Q \setminus R|$ to $T[c + 1 :]$. When we remove a segment from S, we update the tree T in the same way, but we subtract the values.

There is one hurdle that we didn't handle properly: segments that cross the start point 0. We can handle them automatically by inserting a special segment $(-\infty, 0]$ to the set S at the beginning.