

Solutions

GRAD

The problem is a simple graph search on an undirected unweighted graph, where the vertices are the entrance, cranes and destination point. Two vertices are directly connected if it is possible to directly transfer the load from one to another. We need to determine if there is a path from the entrance to the destination in the graph.

KUTIJE

We can order the balls so that the white ones come first or so that the black ones come first. If we can solve both cases optimally, the overall solution will be the cheaper of the two solutions.

Let's solve the first case, the second one is the same with the colors switched. Consider a greedy algorithm which repeats the following 4 steps:

- 1) Proceed from left to right and pick the first two black balls
- 2) Move to the rightmost box that contains a white ball
- 3) Proceed back left, exchanging the black balls for the first white balls we encounter
- 4) Keep going left, filling gaps with white balls

The above algorithm is obviously correct, but also optimal which is less obvious. The proof is complicated, but as a hint, note that we need to fetch the farthest balls at some point so we may as well do it immediately.

KAOS

Let X be the sorted list of words and Y be the sorted list of reversed words.

Now assign two numbers x and y to each word, its position in the lists X and Y .

It is easy to see that the words A and B can't stand each other if $x(A) < x(B)$ and $y(B) < y(A)$.

We process words in order of increasing y and for each word we calculate how many words (of those we've already processed) it cannot stand. Since at any time the current word has the largest y seen so far, we need to find the number of words with a larger x .

To do this efficiently, we need a data structure supporting the following operations:

- 1) Insert integer x
- 2) Find the number of integers larger than x

A one-dimensional segmented array (the "logarithmic structure") is a simple and efficient choice. A two-dimensional variant of this structure appeared in the IOI 2001 task MOBILEs. Another choice is a custom-built binary search tree.