

# Analysis: Saving the Jelly

## Test Set 1

With  $N \leq 10$ , it is possible to use dynamic programming with bitmasking, with one bit for each candy and each student remaining. This effectively lets us try all possible orderings of students as well as all possible ways to break ties. The total complexity is  $O(N^2 \times 2^{2 \times N})$ . This runs in time because we only consider states where the same number of candies and students have been matched.

## Test Set 2

Firstly, let's construct a bipartite graph with the  $N$  children and the  $N$  candies as vertices (not including Mr Jolly's blueberry jelly). Add an edge from child  $a$  to candy  $b$  if child  $a$  is equally close or closer to  $b$  than the blueberry jelly.

It's clear that a necessary (but perhaps not sufficient) condition for Mr Jolly to get his blueberry jelly is that the graph has a [perfect matching](#).

It turns out, this is also a sufficient condition. Let's try to show this by turning a perfect matching into an order that Mr. Jolly can call the children's names.

Firstly, if there is a child who is matched to the candy that is closest to them, then we can call that child's name and remove them and the candy from the graph.

Otherwise, we are in the situation where every child is matched to a (ungrabbed) candy that is *not* the closest one to them. Then, we will find a cycle in the graph using the following procedure. Pick an arbitrary child  $a$  to start at.

1. Find the candy  $b$  that is closest to child  $a$ . Go to  $b$ . This edge is guaranteed not to be part of the current matching.
2. Find the child  $a$  that is currently matched to candy  $b$ . Go to  $a$ . This edge is in the current matching.

Eventually, this process will create a cycle of even length (not necessarily including the child we started with). Because we alternated between edges in the matching and edges not in the matching, we can swap the matched edges/unmatched edges to create a new perfect matching. Note that in doing so, all the children in the cycle are now matched to the closest candy to them, so at least one child's name can now be called.

We have shown that a perfect matching is a necessary condition, and that an order can be constructed from a perfect matching that satisfies Mr. Jolly's requirements, thus proving that a perfect matching exists if and only if Mr. Jolly's requirements can be satisfied.

A perfect matching can be found in  $O(N^2 \sqrt{N})$  using [Hopcroft-Karp](#), and constructing a solution can be done in  $O(N^2)$  if implemented with some care.