# Fairies, heuristics, and A*

In a tranquil garden, five enchanting fairies stand gracefully in a sequence, each adorned with a different magical symbol from 1 to N. These fairies have gathered for a mystical ceremony, and they start in a random order, their wings shimmering in the soft glow of twilight. Their enchanting task is to rearrange themselves in ascending order from 1 to N (e.g., 12345) in as few steps as possible, aligning their mystical symbols in harmony. With a wave of their wands, each fairy can gracefully switch places with only one of her neighboring companions in a single step, creating a mesmerizing spectacle. The gentle sounds of nature and the faint whispers of the wind add to the enchantment of the moment. This whimsical and magical challenge can be seen as a search problem, where the set of possible states $S$ corresponds to all the different configurations of the fairies' symbols (i.e., $S = \{12345, 12354, 12453, ...\}$ for $N = 5$).

We provided a skeleton code to this problem implemented using BFS: solve_fairies.py. You can run it with the command line:

```
python3 solve_fairies.py fairies.txt
```

But this code is very slow. Your job is to convert it into a fast A* solution by modifying the functions h() and solve().

To do this:
1. Switch from a queue to a priority queue.
2. Implement an admissible heuristic $h(s)$.
3. For the priority value, use $f(s) = g(s) + h(s)$, where $g(s)$ is the cost from the initial state to $s$.