For an algorithm which seeks a fast solution, the greedy approach is a great solution and many known algorithms used today follows this principle. In this example we show what a greedy algorithm might look like.

Example:

You are organizing an outdoor skittles tournament where players are attending from same sized groups A and B. Each player has a skill level which is presented by a positive integer. Players must form 2-player teams in such way that one player is from group A and the other player is from group B. In addition the skill gap of teammates must be minimized. Create the teams from given groups A and B such that the sum of skill gaps is minimized. What is the minimum sum of skill gaps?

Unfortunately there is no linear solution for this problem. It may be tempting to search best pairs with nested loops or check all possible pairs but the best and the simplest way to solve this is to get a bit greedy. Idea is to sort both groups by the skill level in increasing order and then pair the players which share same position.

Example 1.1: Greedy solution

```
# pairs.py

# solve best teams, A and B stores the skill levels of the players
def pairs(A, B):
    a = sorted(A)                       # create sorted versions
    b = sorted(B)                       # of A and B (O(n*log(n)))
    res = 0
    for i in range(len(a)):             # Create the teams with
        res += abs(a[i] - b[i])         # a single loop (O(n))
    return res


if __name__ == "__main__":
    print(pairs([7, 4, 5], [3, 6, 2]))
    print(pairs([2, 3, 8, 6, 7], [3, 8, 1, 7, 4]))
    print(pairs([16, 8, 14, 6, 17, 12, 5, 9, 1, 2],
                [7, 18, 10, 2, 1, 3, 8, 4, 13, 19]))
```

Example 1.2: Greedy solution in action

```
3
13
```

At first glance you might suspect that this cannot produce the optimal solution all the time but if you try to minimize the skill gaps inpidually you will be left of pairs that will have a very large skill gap. That is why nested loops does not provide optimal solution all the time. In addition the greedy solution has the running time of $\Theta(n \log n)$ making it also faster than the solution with nested loops ($\Theta(n^2)$).

Last modified: Monday, 3 July 2023, 10:43 AM