Superpowers

Benjamin Schmid

Swiss Olympiad in Informatics

January 7, 2017

Table of Contents

- Good and bad superheroes
- Different strengths s
- $s > 0 \implies$ good
- $s < 0 \implies$ bad
- Device to convert superheroes $s \implies -s$
- Maximize sum of strengths

- Choose C times a superhero
- We can choose the same one multiple times

- We always choose the most negative number (closest to $-\infty$)
- As long as number is negative, $2s$ is added to sum, so we want $|s|$ to be as big as possible
- If no negative number left, swap smallest number to minimize effect on sum
- Implementation
    - sort array
    - left to right swap negative numbers (at most C)
    - if not yet reached C swap the smallest value accordingly

- Choose exactly C different
  superheroes
- Similar to subtask 1
- Sort array
- Swap first C entries

- Swap C continuous heroes
- Equivalent problem: find subarray of length C with minimal sum
- Two approaches: prefix sum (see 2H) or rolling sum

Rolling sum

When you move the window one to the right, one element gets added and one gets subtracted. If we have the sum of the previous window we can calculate the new sum in $\mathcal{O}(1)$.

- Calculate sum of first C elements
- Go through array left to right and update rolling sum
- Store smallest found sum
- Solution is *original_sum* $+ 2|min\_sum|$

- Choose any number of continuous heroes
- Subtask 4 slower solutions
- Problem known as *maximum subarray problem*

- Use solution for subtask 3 for each possible C
- Too slow for subtask 5
- Keep smallest found sum so far and smallest sum ending at current position
- If sum gets positive with current element it's better not to take the sum at all
- Otherwise it's obviously better if we extend the sum
- Solution is again *original_sum* $+ 2|min\_sum|$

```
lli minsum = 0;
lli currentsum = 0;
for (int i = 0; i < n; i++) {
  currentsum += strengths[i];
  if (currentsum > 0) {
    currentsum = 0;
  }
  minsum = min(minsum, currentsum);
}
```

Questions?