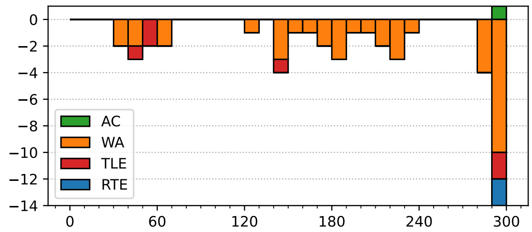
The surface of a spherical cap or a spherical segment is proportional to its height. In cartography, this is known as a cylindrical equal-area projection. Formally, the surface area of any band of height h is $2\pi Rh$ where R is the radius of the sphere. Each horizontal segment covers the same surface area during a whole rotation, and therefore also during a single step. The answer is simply the average of the numbers in the matrix.

Problem J: Bridging the Gap

Problem authors:
Walter Guttmann and Paul Wild

Solved by 1 team.
First solved after 292 minutes.
Shortest team solution: 907 bytes.
Shortest judge solution: 857 bytes.



There are a number of observations that can be made about the structure of the solution:

- Every time a group crosses backwards, the group is only one person
- Every time a group crosses forwards, the group is either:
  - the k slowest people and $c - k$ fastest people who will later cross back, or
  - some k of the fastest people, who will later cross back, or
  - all of the people left, assuming there are no more than c of them.

After every forward crossing except the last, we need to perform one back-crossing. So, we can instead pay for the back-crossing when we cross the fast walkers forward, and keep track of how many paid-for back-crossings we have accumulated.

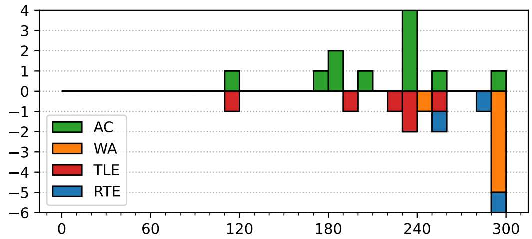
This naturally translates to a dynamic programming solution, where we calculate the cost of having the k slowest people already crossed, and l back-crossings earned. Naively, this is $O(n^3)$ — we have n^2 states, and up to n transitions out of each state. But, if you look closer, it never makes sense to accumulate more that n/c backcrossings (since that's already enough to cross everyone over), so the state space is $O(n^2/c)$. At the same time, the number of transitions out of a state is $O(c)$, so the actual cost after filtering out unreachable states and nonexistent transitions is $O(n^2)$.

While the resulting code is short, a number of judges found the implementation to be tricky to get right.

Problem K: Alea Iacta Est

Problem authors:
Martin Kacer and Arnav Sastry

Solved by 11 teams.
First solved after 113 minutes.
Shortest team solution: 1459 bytes.
Shortest judge solution: 1511 bytes.



This problem can be viewed as a graph problem and solved with either Dijkstra's algorithm or Bellman-Ford. Alternatively, it may be solved with a value iteration algorithm (which, as far as we could tell, was the approach taken by every single team solving this problem). Here we