# Boarding Passes (passes)

**BY MAGNUS HEGDAHL (NORWAY)**

In this task we want to calculate the expected number of times someone passes someone else while boarding according to given boarding groups.

## Subtask 1. $G = 1$

In this subtask there is only one boarding group. Accordingly, we only want to find the expected number of passes for $N$ people boarding in a uniformly random order.

Assume we have $k$ people all boarding from the same entrance. Let $a, b$ be two of these $k$ people with $a$ sitting closer to the entrance. They pass each other if and only if $b$ boards before $a$. The probability of this happening is 0.5, since there are exactly as many permutations where $a$ boards before $b$ as there are permutations with $b$ boarding before $a$. There are $\sum_{i=0}^{k-1} i = \frac{k(k-1)}{2}$ pairs of people each contributing 0.5 to the expected number of passes, for a total of $\frac{k(k-1)}{4}$.

We now want to decide how to assign all $N$ people to the front and back entrances. It is always optimal to choose non-overlapping groups, i.e. we want to find $k_1$ and $k_2 = N - k_1$, where the passengers on the first $k_1$ seats are assigned to the front entrance, and the remaining $k_2$ passengers are assigned to the back entrance, such that the total number of passes is minimal. It is fast enough to find the optimal $k_1$ by linear search, but one may also observe that it is optimal to split in the middle, i.e. $k_1 = \left\lceil \frac{N}{2} \right\rceil$ and $k_2 = \left\lfloor \frac{N}{2} \right\rfloor$. The total number of passes is $\frac{k_1(k_1-1)+k_2(k_2-1)}{4}$.

## Subtask 2. $G \leq 7$, $N \leq 100$

Now we need to find the optimal order of boarding groups. Since $G \leq 7$, it is possible to enumerate all possible orders of boarding groups. We will now consider a fixed order of boarding groups.

For each passenger, the expected number of people they pass while boarding from the front and when boarding from the back can be calculated: For each passenger sitting between the current passenger's seat and the entrance, there is 1 expected pass if the other passenger is in an earlier boarding group, or 0.5 if they are in the same group, and 0 otherwise. This can be calculated for all passengers in $O(N^2)$, which is fast enough for this subtask. It can also be sped up to $O(NG)$ by computing prefix sums. Each passenger is then assigned to the entrance where the expected number of passes is lower.

## Subtask 3. $G \leq 10$, $N \leq 10\,000$

In this subtask, $G \leq 10$. Enumerating all $G! \leq 3\,628\,800$ possible orders could still be possible, but would not leave enough time to calculate the expected number of passes for each order. Instead, a dynamic programming approach is necessary: Consider all $2^G \leq 1024$ subsets of the set of all boarding groups. For each subset, calculate the minimal expected number of passes to board only these groups. This can be calculated as:

$$\mathrm{dp}(S) = \min_{g \in S} \mathrm{dp}(S \setminus \{g\}) + \text{ number of passes while } g \text{ boards after } S \setminus \{g\} \text{ is seated}$$

To calculate $\mathrm{dp}(S)$, first calculate the following values for all positions in $O(N)$:

- How many passengers sitting in front of / behind the current position are part of the same boarding group as the passenger on the current position?
- How many passengers sitting in front of / behind the current position are part of any boarding group $g$ with $g \in S$?

These prefix sums allow us to query in $O(1)$ how many expected passes will be made while a given passenger boards, assuming that all other boarding groups in $S$ have already boarded. Iterate through all groups $g \in S$ and all passengers in group $g$ to find out whether each passenger should board from the front or back entrance, and which group should board last to minimize the expected number of passes.

To calculate this for all groups $g \in S$ and all subsets $S$, we need a running time of $O(2^G N)$, which is sufficient to solve subtask 3. Slightly less efficient solutions in $O(2^G NG)$ are also accepted.

**Subtask 4.** No further constraints.

This subtask requires that we reduce the time needed to calculate the expected number of passes while boarding a group $g$, with a given set of groups already boarded.

For any group $g$, there is a $k^*$ such that in the optimal solution, the first $k^*$ passengers of group $g$ board from the front entrance, and the remaining passengers from the back entrance. Let $p(K)$ be the number of passes if the first $k$ passengers board from the front, and the remaining passengers from the back. The function has a minimum at $k^*$ and is strictly decreasing before the minimum and strictly increasing after.* We can thus use ternary search to find an optimal $k$.

To calculate $p(k)$ for a given $k$ quickly, we precalculate for all groups $g_1, g_2$ and all $1 \le i \le |g_2|$:

- How often is a seat belonging to $g_1$ passed while the first $i$ passengers of group $g_2$ board, assuming they use the front entrance?
- How often is a seat belonging to $g_1$ passed while the last $i$ passengers of group $g_2$ board, assuming they use the back entrance?

We can now calculate $p(k)$ in $O(G)$.

This precalculation can be done in $O(|g_1|)$ for any pair of groups $g_1, g_2$. Thus, to do the precalculation for all pairs of groups, we need time $O(G \sum_{i=1}^{G}(g_i)) = O(GN)$. The ternary search is done for each group in each subset of all groups, i.e. $O(2^G G)$ times. The total running time is thus $O(2^G G^2 + GN)$.

* In some cases there are two values where the minimum is reached.