Official

E - E [max] (/contests/abc411/tasks/abc411_e) Editorial by **en_translator** (/users/en_translator)

Let $S = (S_1, S_2, \dots, S_k)$ be the sequence of integers written on any face of the N dice. For convenience, let $S_0 = 0$.

Let X_i be the random variable for the integer shown by die i. Then the sought expected value E can be written as follows. ($P[*]$ denotes the probability that $*$ happens.)

$$\sum_{i=1}^{k} S_i \times \mathbb{P}[\max_{j=1..N} X_j = k].$$

For a problem asking for a probability or expected value in competitive programming, the following trick can be often applied: rather than finding the probability that some maximum value coincides with a specific value, it is easier to find the probability that the maximum value becomes **less than or equal to** the specific value. This trick can be used for this problem too:

2025-06-21 (Sat)
12:07:59 -04:00

$$
\begin{aligned}
E &= \sum_{i=1}^{k} S_i \times P[\max_j X_j = S_i] \\
&= \sum_{i=1}^{k} S_i \times \left(P[\max_j X_j \le S_i] - P[\max_j X_j \le S_{i-1}] \right) \\
&= S_k - \sum_{i=1}^{k-1} (S_{i+1} - S_i) \times P[\max_j X_j \le S_i] \\
&= S_k - \sum_{i=1}^{k-1} (S_{i+1} - S_i) \prod_{j=1}^{N} \frac{B_i^{(j)}}{6}
\end{aligned}
$$

Here, $B_i^{(j)}$ denotes the number of faces with integers less than or equal to S_i written on them, among the six faces of die j.

We cannot maintain the values $B_i^{(j)}$ for all i and j in an array. However, we can inspect $i = 1, 2, \ldots, k$ in order, and maintain only the values $B_i^{(j)}\ (1 \le j \le N)$ and $\prod_{j=1}^{N} B_i^{(j)}$ for the current i, so that the update operation (that is, setting $B_i^{(j)} \leftarrow B_{i+1}^{(j)}$ for each pair (i, j) with $B_i^{(j)} \ne B_{i+1}^{(j)}$) is done only $O(N)$ time, so this runs fast enough.

The overall time complexity is $O(N(\log N + \log \mathrm{MOD}))$ or $O(N \log N + \log \mathrm{MOD})$ depending on implementation details, but all of them are fast enough.

Sample code (C++):

Copy

2025-06-21 (Sat)
12:07:59 -04:00

```cpp
#include <bits/stdc++.h>
#include <atcoder/modint>

using namespace std;

using mint = atcoder::modint998244353;

int main() {
    int n;
    cin >> n;
    vector<vector<int>> a(n, vector<int>(6));
    vector<int> s;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 6; j++) {
            cin >> a[i][j];
            s.push_back(a[i][j]);
        }
    }
    sort(s.begin(), s.end());
    s.erase(unique(s.begin(), s.end()), s.end());
    int k = s.size();
    vector<vector<int>> upd(k);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 6; j++) {
            int id = lower_bound(s.begin(), s.end(), a[i][j]) - s.begin();
            upd[id].push_back(i);
        }
    }

    mint ans = 0;
    vector<int> b(n);
    mint prod = 1;
    int zero_cnt = n;
    for (int i = 0; i < k - 1; i++) {
        for (int j: upd[i]) {
            if (!b[j]) {
                --zero_cnt;
            } else {
                prod /= b[j];
            }
            ++b[j];
            prod *= b[j];
        }
        ans -= (zero_cnt ? 0 : prod) * (s[i + 1] - s[i]);
    }
    ans /= mint(6).pow(n);
    ans += s[k - 1];
```

2025-06-21 (Sat)
12:07:59 -04:00

```
    cout << ans.val() << endl;
}
```

posted: about 11 hours ago

last update: about 11 hours ago

am)

ps%3A%2F%2Fatcoder.jp%2Fcontests%2Fabc411%2Feditorial%2F13369%3Flang%3Den&title=Editorial%20-
g%20Contest%202025%20Summer%20(AtCoder%20Beginner%20Contest%20411))

2025-06-21 (Sat)
12:07:59 -04:00