

Contest Duration: 2025-06-21(Sat) 08:00 (<http://www.timeanddate.com/worldclock/fixedtime.html?iso=20250621T2100&p1=248>) - 2025-06-21(Sat) 09:40 (<http://www.timeanddate.com/worldclock/fixedtime.html?iso=20250621T2240&p1=248>) (local time) (100 minutes)

iso=20250621T2100&p1=248) - 2025-06-21(Sat) 09:40 (<http://www.timeanddate.com/worldclock/fixedtime.html?iso=20250621T2240&p1=248>) (local time) (100 minutes)

iso=20250621T2240&p1=248) (local time) (100 minutes)

[Back to Home \(/home\)](#)

[🏠 Top \(/contests/abc411\)](#) [☰ Tasks \(/contests/abc411/tasks\)](#)

[❓ Clarifications \(/contests/abc411/clarifications\)](#) [🚀 Submit \(/contests/abc411/submit\)](#)

[☰ Results ▾](#) [⬇️ Standings \(/contests/abc411/standings\)](#)

[⬇️ Virtual Standings \(/contests/abc411/standings/virtual\)](#)

[🔧 Custom Test \(/contests/abc411/custom\\_test\)](#)

[📖 Editorial \(/contests/abc411/editorial\)](#)

[💬 Discuss \(<https://codeforces.com/blog/entry/143934>\)](#)



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

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

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  $i = 1, 2, \dots, k$  in order, and maintain only the values  $B_i^{(j)}$  ( $1 \leq j \leq N$ ) 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  $B_i^{(j)} \neq B_{i+1}^{(j)}$ ) is done only  $O(N)$  time, so this runs fast enough.

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

Sample code (C++):

Copy

```
1. #include <bits/stdc++.h>
2. #include <atcoder/modint>
3.
4. using namespace std;
5.
6. using mint = atcoder::modint998244353;
7.
8. int main() {
9.     int n;
10.    cin >> n;
11.    vector<vector<int>> a(n, vector<int>(6));
12.    vector<int> s;
13.    for (int i = 0; i < n; i++) {
14.        for (int j = 0; j < 6; j++) {
15.            cin >> a[i][j];
16.            s.push_back(a[i][j]);
17.        }
18.    }
19.    sort(s.begin(), s.end());
20.    s.erase(unique(s.begin(), s.end()), s.end());
21.    int k = s.size();
22.    vector<vector<int>> upd(k);
23.    for (int i = 0; i < n; i++) {
24.        for (int j = 0; j < 6; j++) {
25.            int id = lower_bound(s.begin(), s.end(), a[i][j]) - s.begin();
26.            upd[id].push_back(i);
27.        }
28.    }
29.
30.    mint ans = 0;
31.    vector<int> b(n);
32.    mint prod = 1;
33.    int zero_cnt = n;
34.    for (int i = 0; i < k - 1; i++) {
35.        for (int j: upd[i]) {
36.            if (!b[j]) {
37.                --zero_cnt;
38.            } else {
39.                prod /= b[j];
40.            }
41.            ++b[j];
42.            prod *= b[j];
43.        }
44.        ans -= (zero_cnt ? 0 : prod) * (s[i + 1] - s[i]);
45.    }
46.    ans /= mint(6).pow(n);
47.    ans += s[k - 1];
```

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

```
48.     cout << ans.val() << endl;  
49. }
```

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))

---

[Rule \(/contests/abc411/rules\)](/contests/abc411/rules) [Glossary \(/contests/abc411/glossary\)](/contests/abc411/glossary)

[Terms of service \(/tos\)](/tos) [Privacy Policy \(/privacy\)](/privacy) [Information Protection Policy \(/personal\)](/personal) [Company \(/company\)](/company)

[FAQ \(/faq\)](/faq) [Contact \(/contact\)](/contact)

Copyright Since 2012 ©AtCoder Inc. (<http://atcoder.co.jp>) All rights reserved.

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