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)

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

D - Conflict 2 (/contests/abc411/tasks/abc411_d) Editorial by en_translator (/users/en_translator)

Copying a string of length $O(N)$ costs $O(N)$ time, so you cannot naively simulate the procedure specified by each query. One could use a data structure like a persistent array, but here we will introduce a solution with simpler implementation.

For simplicity, we call the server PC 0.

Let $f(t,i)\ (0 \leq t \leq Q, 0 \leq i \leq N)$ be the string held by PC i after processing the first t queries.

Then the operations for the queries of each kind can be translated using f as follows:

- For each $t\ (1 \leq t \leq Q)$, if the t-th query is
  - `1 p` : $f(t,p) = f(t-1,0)$, $f(t,i) = f(t-1,i)\ (i \neq p)$
  - `2 p s` : $f(t,p) = f(t-1,p) + s$, $f(t,i) = f(t-1,i)\ (i \neq p)$ (where $+$ denotes string concatenation)
  - `3 p` : $f(t,0) = f(t-1,p)$, $f(t,i) = f(t-1,i)\ (i \neq 0)$

What we want is $f(Q,0)$, so we can apply the relations above repeatedly for $t = Q, Q-1, \ldots, 1$ in order (in other words, by scanning the queries in reverse order), the problem is reduced to finding $f(0,i)$ for a certain i. Since we know that this is nothing but an empty string, so the answer is found.

Specifically, the algorithm is described as follows:

1. Let $t = Q, i = 0, \mathrm{ans} =$ (an empty string).
2. If $\mathrm{query}_t =$ `1 p` and $i = p$, set $i \leftarrow 0$.
3. If $\mathrm{query}_t =$ `2 p s` and $i = p$, set $\mathrm{ans} \leftarrow s + \mathrm{ans}$.

2025-06-21 (Sat)
10:20:22 -04:00

4. If $\mathrm{query}_t =$ `3 p` and $i = 0$, set $i \leftarrow p$.
5. Set $t \leftarrow t - 1$.
6. Terminate the algorithm if $t = 0$, and go back to step 2. otherwise.

This algorithm runs in $O(Q + \sum |s|)$ time.

Implementation note: In many languages like C++ and Python, simply writing like `ans = s + ans` , will impose an additional cost of copying `ans` , worsening the time complexity. Instead of managing `ans` itself, one can manage the reversed string `rev(ans)` of `ans` (in this case, step 3. is achieved by `rev(ans) += rev(s)`), and finally reverse the string again to obtain `ans` .

Sample code (C++):

Copy

2025-06-21 (Sat)
10:20:22 -04:00

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

using namespace std;

int main() {
    int n, q;
    cin >> n >> q;
    vector<int> op(q), p(q);
    vector<string> s(q);
    for (int i = 0; i < q; i++) {
        cin >> op[i] >> p[i];
        if (op[i] == 2) {
            cin >> s[i];
            reverse(s[i].begin(), s[i].end());
        }
    }

    string ans;
    int i = 0;
    for (int t = q - 1; t >= 0; t--) {
        if (op[t] == 1) {
            if (i == p[t]) {
                i = 0;
            }
        } else if (op[t] == 2) {
            if (i == p[t]) {
                ans += s[t];
            }
        } else {
            if (i == 0) {
                i = p[t];
            }
        }
    }
    reverse(ans.begin(), ans.end());
    cout << ans << endl;
}
```

posted: about 12 hours ago

last update: about 12 hours ago

am)

ps%3A%2F%2Fatcoder.jp%2Fcontests%2Fabc411%2Feditorial%2F13367%3Flang%3Den&title=Editorial%20-
g%20Contest%202025%20Summer%20(AtCoder%20Beginner%20Contest%20411))

2025-06-21 (Sat)
10:20:22 -04:00

2025-06-21 (Sat)
10:20:22 -04:00