

# B-正义切割

记 $dp(i)$ 为对字符串 $S[0 : i]$ 切割的价值总和

$num(i, j)$ 位字符串 $S[i : j]$ 对应的数字

有 $num(i, j) = num(i, j - 1) * 10 + num(j - 1, j)$

考虑第 $i$ 位所在数字的长度, 可以推导出 $i \geq 2$ 时

$$\begin{aligned} dp(i) &= \sum_{j=0}^{i-1} dp(j) * num(j, i) \\ &= dp(i-1) * num(i-1, i) + \sum_{j=0}^{i-2} dp(j) * num(j, i) \\ &= dp(i-1) * num(i-1, i) + \sum_{j=0}^{i-2} dp(j) * [num(j, i-1) * 10 + num(i-1, i)] \\ &= \sum_{j=0}^{i-1} dp(j) * num(i-1, i) + 10 \sum_{j=0}^{i-2} dp(j) * num(j, i-1) \\ &= num(i-1, i) \sum_{j=0}^{i-1} dp(j) + 10 dp(i-1) \end{aligned}$$

其中 $\sum_{j=0}^{i-1} dp(j)$ 可以边算边求, 总的时间复杂度为 $O(|S|)$ .

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

using namespace std;

typedef long long ll;

ll t, n;
ll MOD = 998244353;
ll dp[100005];

string s;
int main() {
    cin >> t;
    while (t--) {
        cin >> s;
        n = s.length();
        s = "-" + s;
        dp[0] = 1;
        dp[1] = s[1] - '0';
        ll tot = 0;
        for (int i = 2; i <= n; ++i) {
            int si = s[i] - '0';
            dp[i] = dp[i-1] * (10 + si) % MOD;
            tot += dp[i-2];
            tot %= MOD;
            dp[i] += si * tot % MOD;
            dp[i] %= MOD;
        }
    }
}
```

```
        cout << dp[n] << endl;  
    }  
    return 0;  
}
```