

目次

1 文字列

1.1 Aho-Corasick 法

1 文字列

1.1 Aho-Corasick 法

$O(N + M)$

```
1 struct PMA{
2     PMA* next[256];    //0 is failure link
3     vector<int> matched;
4     PMA(){memset(next, 0, sizeof(next));}
5     ~PMA(){for(int i = 0; i < 256; i++) if(next[i]) delete next[i];}
6 };
7 vector<int> set_union(const vector<int> &a, const vector<int> &b){
8     vector<int> res;
9     set_union(all(a), all(b), back_inserter(res));
10    return res;
11 }
12 // patternからパターンマッチングオートマトンの生成
13 PMA *buildPMA(vector<string> pattern){
14     PMA *root = new PMA, *now;
15     root->next[0] = root;
16     for(int i = 0; i < pattern.size(); i++){
17         now = root;
18         for(int j = 0; j < pattern[i].size(); j++){
19             if(now->next[(int)pattern[i][j]] == 0)
20                 now->next[(int)pattern[i][j]] = new PMA;
21             now = now->next[(int)pattern[i][j]];
22         }
23         now->matched.push_back(i);
24     }
25     queue<PMA*> que;
26     for(int i = 1; i < 256; i++){
27         if(!root->next[i]) root->next[i] = root;
28         else {
29             root->next[i]->next[0] = root;
30             que.push(root->next[i]);
31         }
32     }
33     while(!que.empty()){
34         now = que.front(); que.pop();
35         for(int i = 1; i < 256; i++){
36             if(now->next[i]){
37                 PMA *next = now->next[i];
38                 while(!next->next[i]) next = next->next[0];
39                 now->next[i]->next[0] = next->next[i];
40                 now->next[i]->matched = set_union(now->next[i]->matched, next->next[i]->matched);
41                 que.push(now->next[i]);
42             }
43         }
44     }
45     return root;
46 }
47 void match(PMA* &pma, const string s, vector<int> &res){
48     for(int i = 0; i < s.size(); i++){
49         int c = s[i];
50         while(!pma->next[c])
51             pma = pma->next[0];
52         pma = pma->next[c];
53         for(int j = 0; j < pma->matched.size(); j++){
54             res[pma->matched[j]] = true;
55         }
56     }
```