

```
1  /*
2   | Trie |
3   Desc: Multiple string matching in O(max(s.size())) for all operations
4   Source: KawakiMeido
5   State: VERY Untested and old code lmao
6  */
7
8  struct Trie{
9
10     struct Node{
11         Node* child[2];
12         int cnt;
13         int l,r;
14         Node(){
15             child[0] = child[1] = NULL;
16             cnt = 0;
17         }
18     };
19
20     Node* Root;
21     int cnt;
22     Trie(){
23         Root = new Node();
24         cnt = 0;
25     }
26
27     void Init(){
28         clr(Root);
29     }
30     void clr(Node* cur){
31         if (cur->child[0] != NULL){
32             clr(cur->child[0]);
33             cur->child[0] = NULL;
34         }
35         if (cur->child[1] != NULL){
36             clr(cur->child[1]);
37             cur->child[1] = NULL;
38         }
39         if (cur != Root) delete cur;
40     }
41
42     void Add(int x, int pos){
43         Node* cur = Root;
44         for (int i=29; i >= 0; i--){
45             int idx = ((x>>i)&1);
46             if (cur->child[idx] == NULL) cur->child[idx] = new Node();
47             cur = cur->child[idx];
48             cur->cnt++;
49         }
```

```
50     }
51     int Get(int x, Node* cur, int lg){
52         int res = 0;
53         if (cur->cnt ≤ 0) return INF;
54         for (int i=lg; i ≥ 0; i--){
55             int idx = ((x>>i)&1);
56             if (cur->child[idx] = NULL){
57                 res = res+(1<<i);
58                 cur = cur->child[(idx+1)%2];
59             }
60             else{
61                 cur = cur->child[idx];
62             }
63         }
64         return res;
65     }
66 };
67
68 Trie TR;
```