**Aho-Corasick:** Builds the Aho-Corasick automaton.

```cpp
template<int K = 26> class AhoCorasick {
  struct Node {
    Node* tr[K];       // transitions
    Node* suff;        // dictionary suffix
    vector<Node*> adj; // incoming dict suffixes

    Node() : suff(nullptr) {
      fill(tr, tr + K, nullptr);
    }
  };

  Node* root;
  vector<Node*> dict;

  Node* insert(const string &s) {
    Node* curr = root;
    for (auto c: s) {
      if (!curr->tr[c - 'a'])
        curr->tr[c - 'a'] = new Node;
      curr = curr->tr[c - 'a'];
    }

    return curr;
  }

  void get_suffixes() {
    queue<Node*> q;

    for (int i = 0; i < K; i++) {
      if (root->tr[i]) {
        root->tr[i]->suff = root;
        root->adj.push_back(root->tr[i]);
        q.push(root->tr[i]);
      } else {
        root->tr[i] = root;
      }
    }

    while (!q.empty()) {
      Node* curr = q.front(); q.pop();

      for (int i = 0; i < K; i++) {
        if (curr->tr[i]) {
          curr->tr[i]->suff = curr->suff->tr[i];
          curr->tr[i]->suff->adj
            .push_back(curr->tr[i]);
          q.push(curr->tr[i]);
        } else {
          curr->tr[i] = curr->suff->tr[i];
        }
      }
    }
  }

public:

  AhoCorasick(const vector<string> &words) {
    root = new Node;
    for (auto &word: words) {
      dict.push_back(insert(word));
    }
    get_suffixes();
  }
};
```