アルゴリズムとデータ構造B 成果発表会（G9）

メンバー

- 水本 幸希
- 山口 慧
- 杣谷 星音
- 斧田 洋人
- 杉山 亮太

基本コンセプト

与えられたクエリと放送局のデータを切り取ったもの比較して、編集距離が最短となった放送局を答えとする。

編集距離の求め方

編集距離を求める方法には様々なものが存在するが、今回はビットパラレル法を採用した。
ビットパラレル法は動的計画法を応用しており、ビット演算を利用することで並列処理を行うことで計算時間を短縮している。
動的計画法の場合、計算量は$O(N^2)$であるが、ビットパラレル法を用いると$O(N)$まで計算量を減らすことができる。ただし、この計算量にするには配列の要素数が演算に使用するビット長と同じかそれ以下でなければならない。C言語の場合、使用できる変数の最大ビット長は64ビットであるからクエリの長さが65文字以上だとそのまま適用することができない。そのため今回は**クエリの65文字目以降を捨てる**ことで解決させた。これはクエリの長さが長いほど正答率が高くなり、64文字あればほぼ確実に正解を出せるためである。

各アルゴリズムでの計測時間の比較

	動的計画法	O(ND)アルゴリズム	O(NP)アルゴリズム	ビットパラレル法
実行時間（sec）	40秒前後	8秒前後	4秒前後	2秒前後

今回調べた中ではビットパラレル法が最も有効な手法であると考え、採用した。

bitparallel weighted Levenshtein distance
https://stackoverflow.com/questions/65363769/bitparallel-weighted-levenshtein-distance

放送局の信号分割と打ち切りの条件

信号をクエリの長さだけ切り取る、ということを全ての箇所で行えば理論上かなり高い確率で当てることができる。ただし、切り取った開始位置の候補は、50万個存在しこれを全て走査して10秒以内に終わらせることは現実的ではない。そのため今回は、放送局の持つ信号からクエリの長さ分だけ信号を切り取って（最初は当然放送局の最初の位置から）**編集距離を調べた後、放送局から切り取った信号の開始地点から$(\text{クエリの長さ})/10$だけ右に移動**し、再度編集距離を求める、といったことを繰り返すということを繰り返した。この場合、切り取った部分を走査するときは最大で$1/20$ずれる可能性がある。しかし、これ以上細かく走査してもスコアは上がらず、逆に粗くすると正答率が低くなったので、走査するときのステップはこのように設定した。

なお、これだけでも十分時間内に答えを求めることができるが、編集距離が著しく低いものが発見されたらその時点で答えを確定させて打ち切ってしまえば、正答率を下げることなく必要な時間をさらに短縮することができる。具体的には、**編集距離がクエリの長さの$1/4$以下になった場合はその場で打ち切っている**。これ以上小さくしても結果は変わらず逆に大きくすると、間違える可能性が増加した。

クエリを聞き直す条件

これまでの工夫で、それなりに正解を出すことができるが信号が短い場合の正答率が低くなってしまう。これは各放送局での最短編集距離が最も小さい放送局が複数存在する場合、最初に発見した放送局を答えとしてしまうようなプログラムにしたことが原因であった。これを解決するため、最短編集距離が最も短いものが複数存在する場合はクエリを聞き直し、それらの放送局に対して最短編集距離を再度求めることで、放送局を確定させた。これによってさらに正答率を上昇させることができた。

苦戦したところ

今回ビットパラレル法を用いた手法を提案した。この手法では**挿入、削除コストは1、置換コストは2**となっている（置換は考慮されず、挿入と削除を組み合わせて計算しているため）。色々試したところ、今回の条件では置換コストを2とした方が確実に正答率が上昇したため置換コストは1とすることはしなかった。そのため、挿入、削除率が高い場合でも高いスコアを出すことができるが、**置換率が高い状況では高いスコアを出すことが難しかった**。特に、クエリの長さが短い場合は正答を出すことが困難であり今回はそれらのケースで確実に正解を出すことは断念した。

また、メモリピークを抑えるために各データを2ビット（0, 1, 2, 3）に圧縮しようとも考えたが、実行時間が著しく長くなるため、これも却下となった。

結果

中間計測とほぼ同じプログラム（多少変えている部分はあるが）を提出したので中間計測のスコア（平均値）を記載する。（下の二つは最初に配布された10個のテストケースでの平均値）

- スコア　9705.3
- 実行時間　1.39　[sec]
- メモリピーク　2184.48 [KB]
- クエリを聞き直した回数　9.7 回
- 編集距離が短いものを発見して中断した回数　62.7 回

答えを外したのは、クエリが短かったり、置換率が高い場合がほとんどであった。

まとめ

- ビットパラレル法を用いて各放送局の最短編集距離を求める
- 走査は1バイトずつではなく、1回につきクエリの長さの$1/10$だけ右に進めていく
- 編集距離が著しく短い（クエリの長さの$1/4$以下）場合はその時点で答えを確定させてそのクエリに対する走査は打ち切る
- 最短編集距離が複数で同一だった場合はクエリを聞き直して再度求めなおす
- 置換コストは2、挿入、削除コストを1とすると切り取った部分以外との編集距離を増やすため正確に答えを出せる
- 今回の手法では、「クエリが短い」、「置換率が高い」場合に正答率が低くなる

実行フローチャート