C: Cyclic String

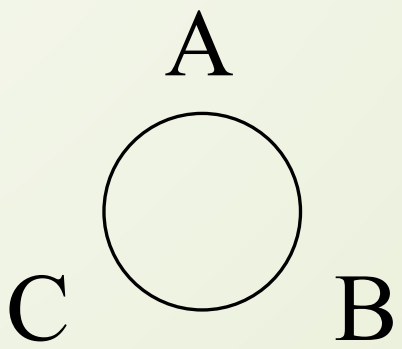
原案: potetisensei

解法: potetisensei

テスト: oyas

問題

- 文字列S_1, S_2, … , S_Nが与えられる。これらを”巡回的に“含むような文字列集合であって、合計文字列長が最小になるものを求めよ(ただし文字列集合の大きさは問わない)



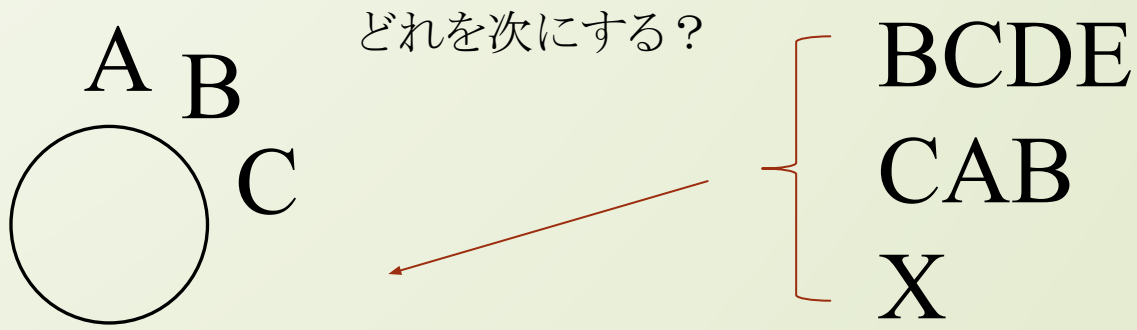
ABCAB

考察

- 求める文字列集合に、明らかに無駄な文字を入れる必要はない
  - 例えば与えられた文字列にzが含まれていないのに、解にzを入れる必要はない
  - もっといえば、文字種に限らず、解に含まれている文字は常に“条件を満たすために”含めるべきである
  - したがって、解は与えられた文字列の部分文字列をつなげたものになっているはず
    - より厳密には、うまく切り分けると、解は必ず入力の接尾辞 /接頭辞のみからなる
- 入力の各文字列は、必ずどこかに含まれていないといけない一方、2度含める理由はない
  - 構成した結果2度含まれることはあるかもしれないが、意図的に含めるのは1度で良い

考察

- 円環に文字列を“埋め込む”事を考える
  - さきほどの考察から、ある文字列を円環に埋め込んだ時、その文字列に続くのは必ず他の文字列の接尾辞。あるいは、そこで円環を閉じるかの2択
    - もちろん、埋め込んだ文字列の接尾辞とその次に埋め込む接尾辞で、入力の文字列は構成されていなければならない
  - したがって、「ある円環のある場所に自分自身を埋め込むことにして、次にどの文字列を埋め込むか？（自分自身であっても良い）」について考えることにする。
    - 自分自身の接尾辞と、次に埋め込む文字列の接頭辞が長く一致するほど、次に埋め込む接尾辞は短くなる

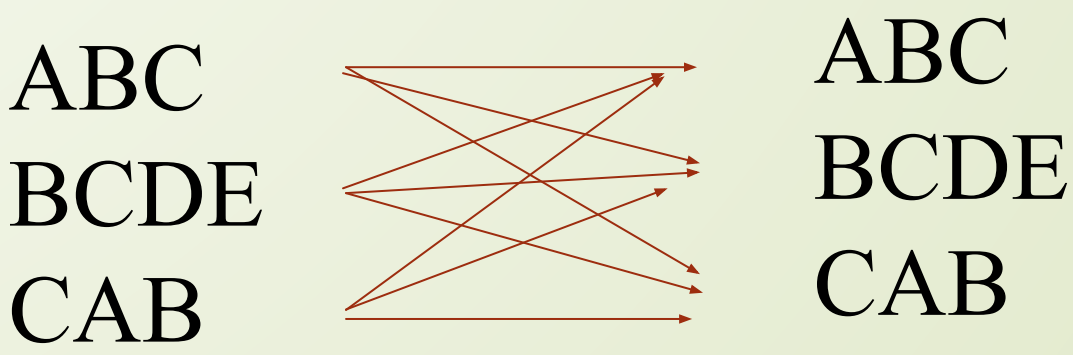


考察

- この、「自分の後に何が来るべきか？」を考えるのは置換pを定めることに等しい。
  - 文字列 S_i の次に文字列 S_{p(i)} を最大限重ねて置くことにすればよい（最大限に重ねない理由はない）
  - 結局出来上がる解の合計文字列長は、この際どれだけ多く重ねられるかで決まる
- ところで、このような状況下で使えるアルゴリズムがありましたね？

最小費用流

- 二部グラフ上で、頂点 u_i から v_j に対して、「文字列S_i から 文字列 S_j につなげることにすると何文字重なるか」をコストとして辺を張る
  - 自分自身にも辺を貼る。ただし、自分自身に貼る時は、文字列全てを重ねないようにする
- こうして出来たグラフ上で最大完全マッチングを求めればよい
  - 完全マッチングなので、適当な大きい定数からコストを引いたグラフで最小費用流



貪欲

- 最小費用流は解き方としては非常に綺麗だが、実は最小費用流は必要ない
  - 実装コストが大きいので、ICPCには向かない
- 実はこの問題は貪欲で解くことが出来る
  - マッチングのコストが特殊
- 与えられた文字列に対して、最も重なるものを順につなげていけばよい
  - 入力される文字列には、一方がもう一方に真に含まれるケースはないので、接尾辞と接頭辞の重なる長さは更新されない
  - したがって比較的簡単な実装によってO(n^2 max(|S_i|))などで求める事が出来る

ちなみに

- この問題はshortestsuperstring問題というものに関連している
  - 4近似アルゴリズムなどでは、この問題を部分問題として用いる
    - 入力された文字列をグラフとして捉えると、実は TSPに対するminimum cycle coverのような存在
- O(n)で解くことが出来る事も知られている
  - Arxivかどこかに上がっていた気がします