

Strings

Suffix Array – Aplicações

Prof. Edson Alves - UnB/FGA

2019

1. Busca em *array* de sufixos

Busca em *array* de sufixos

Busca em *array* de sufixos

- O problema de se determinar se a string P , de tamanho M , é ou não uma substring de S , de tamanho N , pode ser resolvido por meio de um *array* de sufixos
- Isto porque, se P é uma substring de S , ela será substring de algum dos prefixos $S[i..N]$ de S
- Assim, para localizar P em S basta fazer uma busca binária em $s_A(S)$
- Em cada etapa da busca binária, a comparação de T com o prefixo em questão tem complexidade $O(M)$
- Assim o algoritmo tem complexidade $O(M \log N)$, se $s_A(S)$ já estiver construído
- O número de ocorrências de P pode ser determinado por meio de uma segunda busca binária, pois todas elas estarão adjacentes no *array* de sufixos

Implementação da busca em *array* de sufixos em C++

```
84 int occurrences(const string& P, const string& S)
85 {
86     auto sa = suffix_array(S);
87
88     auto it = lower_bound(sa.begin(), sa.end(), P,
89         // retorna true S[sa[i]..N] de anteceder P na ordenação
90         [&](int i, const string& P) {
91             return S.compare(i, P.size(), P) < 0;
92         });
93
94     auto jt = upper_bound(sa.begin(), sa.end(), P,
95         // retorna true se P deve anteceder S[sa[i]..N] na ordenação
96         [&](const string& P, int i) {
97             return S.compare(i, P.size(), P) > 0;
98         });
99
100     return jt - it;
101 }
```

1. CP Algorithms. [Suffix Array](#), acesso em 06/09/2019.
2. **CROCHEMORE**, Maxime; **RYTTER**, Wojciech. *Jewels of Stringology: Text Algorithms*, WSPC, 2002.
3. **HALIM**, Steve; **HALIM**, Felix. *Competitive Programming 3*, Lulu, 2013.