

# suffix-arrays

September 19, 2020

## 0.1 Suffix arrays

Let's create a suffix array of the string "sacramento"

```
In [1]: s = "sacramento"
```

```
In [2]: suffixes = [s[i:] for i in range(len(s))]  
suffixes
```

```
Out[2]: ['sacramento',  
         'acramento',  
         'cramento',  
         'ramento',  
         'amento',  
         'mento',  
         'ento',  
         'nto',  
         'to',  
         'o']
```

```
In [3]: suffix_arr = sorted(suffixes)  
suffix_arr
```

```
Out[3]: ['acramento',  
         'amento',  
         'cramento',  
         'ento',  
         'mento',  
         'nto',  
         'o',  
         'ramento',  
         'sacramento',  
         'to']
```

```
In [4]: indices = [(i, len(s)) for i in range(len(s))]  
indices
```

```

Out[4]: [(0, 10),
          (1, 10),
          (2, 10),
          (3, 10),
          (4, 10),
          (5, 10),
          (6, 10),
          (7, 10),
          (8, 10),
          (9, 10)]

In [5]: suffix_arr = sorted(indices, key=lambda t: s[t[0]:t[1]])
suffix_arr

Out[5]: [(1, 10),
          (4, 10),
          (2, 10),
          (6, 10),
          (5, 10),
          (7, 10),
          (9, 10),
          (3, 10),
          (0, 10),
          (8, 10)]

```

With these much more memory efficient indices, we are able to convert them into strings when needed

```

In [6]: indices_to_suffixes = [s[i:j] for i, j in suffix_arr]
indices_to_suffixes

Out[6]: ['acramento',
          'amento',
          'ramento',
          'ento',
          'mento',
          'nto',
          'o',
          'ramento',
          'sacramento',
          'to']

```

You never want to convert the entire array though, but having a function that returns the string when needed is fine

```

In [7]: from typing import Tuple
def get_suffix(string: str, indices: Tuple[int, int]):
    l, r = indices
    return string[l:r]

```

```
In [8]: s1 = "Sacramento is a city in california"
s2 = "I was there when i was six"
pp1 = s1.lower().split()
pp2 = s2.lower().split()
doc = pp1 + pp2
sa = sorted([" ".join(pp1[i:]) for i in range(len(pp1))]) + [" ".join(pp2[i:]) for i in range(len(pp2))]
sa
```

```
Out[8]: ['a city in california',
 'california',
 'city in california',
 'i was six',
 'i was there when i was six',
 'in california',
 'is a city in california',
 'sacramento is a city in california',
 'six',
 'there when i was six',
 'was six',
 'was there when i was six',
 'when i was six']
```

```
In [9]: sa = [(i, len(pp1)) for i in range(len(pp1))] + [(i, len(pp1)+len(pp2)) for i in range(len(pp2))]
sa
```

```
Out[9]: [(0, 6),
 (1, 6),
 (2, 6),
 (3, 6),
 (4, 6),
 (5, 6),
 (6, 13),
 (7, 13),
 (8, 13),
 (9, 13),
 (10, 13),
 (11, 13),
 (12, 13)]
```

```
In [10]: sorted(sa, key=lambda t: " ".join(doc[t[0]:t[1]]))
```

```
Out[10]: [(2, 6),
 (5, 6),
 (3, 6),
 (10, 13),
 (6, 13),
 (4, 6),
 (1, 6),
 (0, 6),
```

(12, 13),  
(8, 13),  
(11, 13),  
(7, 13),  
(9, 13)]