COT 5405: Fall 2006

Lecture 25

KMP Algorithm for String matching

Prefix function

$\pi(q): \{1, \ldots, m\} \rightarrow \{0, \ldots, m-1\} = \max\{k: k < q \text{ and } P_k \text{ is a suffix of } P_q\}$.

```
Compute π(P)

• π[1] ← 0
• k ← 0
• for q = 2 to m
    o while k > 0 and P[k+1] ≠ P[q]
        ▪ k ← π[k]        ← Withdraw
    o if P[k+1] == P[q]
        ▪ k ← k+1         ← Pay twice
    o π[q] ← k
• return π
```

This takes $\Theta(m)$ time. This can be shown through amortized time complexity analysis.

KMP Algorithm

```
KMP(T, P)

• π ← Compute π(P)
• q ← 0
• for i = 1 to n
    o while q > 0 and P[q+1] ≠ T[i]
        ▪ q ← π[q]
    o if P[q+1] == T[i]
        ▪ q ← q+1
    o if q == m
        ▪ Print i-m
        ▪ q ← π[q]
```

Correctness and time complexity analysis are similar to that of π.