Lecture 3

Aho's survey

- See what topics there are in string algorithms.
- Read it as lectures cover the material.
- References

String Matching

Input: pattern $P[1..m]$, text $T[1..n]$

1. Preprocess pattern P, and produce something (finite automata)
2. Search text T

Algorithms better than $O(mn)$ time in the worst case and $O(n)$ time in the average case?

Karp-Miller-Rosenberg: $O(n \log m)$ time

Knuth-Morris-Pratt Algorithm

Examples

```
T: a b a c b a b a b a a b c b a b
P:           a b a b a c a
```

shift 2 positions

```
T: b a c a b c a b c d a b c
P:       a b c a b c a
```

shift 3 (Prefix) or more (real KMP)

Prefix Function

Let $P_q = P[1..q]$.

Prefix function (failure function) $f(q)$, $1 \leq q \leq m$, is the length of the longest proper suffix of P_q which is a prefix of P_q.

Examples of prefix function

```
a b a b a c a
0 0 1 2 3 0 1
```

```
a a b b a a b
0 1 0 0 1 2 3
```

Pattern-Matching Machine

Pattern-matching machine for KMP: See p331 of AHU. It's like a finite automaton.

```
  a   a   b   b   a   a   b
0   1   2   3   4   5   6   7
```


KMP Text Search

Idea:

- Compare T and P from left.
- Whenever there is a mismatch, the next possible occurrence is the prefix-suffix of the matched part.

Figure

Example:

```
T  ...abaaababaaabaa...
P     abaaababaaabac
            abaaabab..
                abaa..
```

```
procedure KMP(T,P)
  f = Prefix(P)
  q = 0  (current state of pattern matching machine)
  for i = 1 to n do
    while q > 0 and P[q+1] != T[i] do
      q = f(q)
    od
    if q > 0 or P[q+1] == T[i] then q = q+1 fi
    if q == m then
        print "occurrence at i-m"
        q = f(q)
    fi
  od
end
```


KMP Pattern Processing

Idea:

- $f(1) = 0$ always. Compute $f(2)$ to $f(m)$.
- Assume that $f(1), \ldots, f(q-1)$ have been computed. Now compute $f(q)$. Let $k = f(q-1)$. If $P[k+1] = P[q]$ then $f(q) = f(q-1) + 1$ else repeat with $k = f(k)$.

Figure

```
   procedure Prefix(P)
1    f(1) = 0
2    k = 0
3    for q = 2 to m do
4      while k > 0 and P[k+1] != P[q] do
5        k = f(k)
6      od
7      if k > 0 or P[k+1] == P[q] then k = k+1 fi
8      f(q) = k
9    od
10   return f
   end
```


Running Time

Running time of Prefix

- constant time: steps 1,2,10
- $O(m)$: steps 3,7,8,9

How to count while loop: Count how many times step 5 is executed.

Look at k.

- Initially $k = 0$.
- At the end $k < m$.
- k increases by 1 in step 7.
- k decreases each time step 5 is executed (so if $k > 0$).
- The number of increases by 1 is at most m.
- What's the number of decreases? At most m.

Push and Multi-Pop operations. So step 5 is executed at most m times, and while loop at most $2m$ times.

Running time of KMP: similar to Prefix.

Look at q in while loop

- q increases by 1 at most n times
- The number of decreases is at most n.