**Naive string matching**:

for (i=0; T[i] != '\0'; i++)

{

for (j=0; T[i+j] != '\0' && P[j] != '\0' && T[i+j]==P[j]; j++) ;

if (P[j] == '\0') found a match

}

strcpy(s,t)

strcat(s,t)

strcmp(s,t)

Strnins(s, t,i)

If(i<0 && i>strlen(s))

Error

Strncpy(temp,s,i)

Srtcat(temp,t)

Strcat(temp, (s+i))

Strcpy(s,temp)

Failure function

Longest prefix and suffix

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | A | B | E | A | B | F |
| -1 | -1 | -1 | -1 | 0 | 1 | -1 | 0 | 1 | -1 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | A | B | F | A | B | C |
| -1 | -1 | -1 | -1 | -1 | 0 | 1 | -1 | 0 | 1 | 2 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | A | B | C | A | D | A | A | B | E |  |
| -1 | 0 | -1 | -1 | 0 | -1 | 0 | 1 | 2 | -1 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | A | A | A | B | A | A | C | D |
| -1 | 0 | 1 | 2 | -1 | 0 | 1 | -1 | -1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | A | B | C | A | C | A | B |
| -1 | -1 | -1 | 0 | 1 | 2 | 3 | -1 | 0 | 1 |

Pat =”abcabcacab”

Fail(char \*pat){

failure[0]=-1

n=strlen(pat)

for(j=1 ; j<n; j++){

i=failure[j-1]

while(pat[j]!=pat[i+1])) && i>=0)

i=failure[i]

if(pat[j]==pat[i+1])

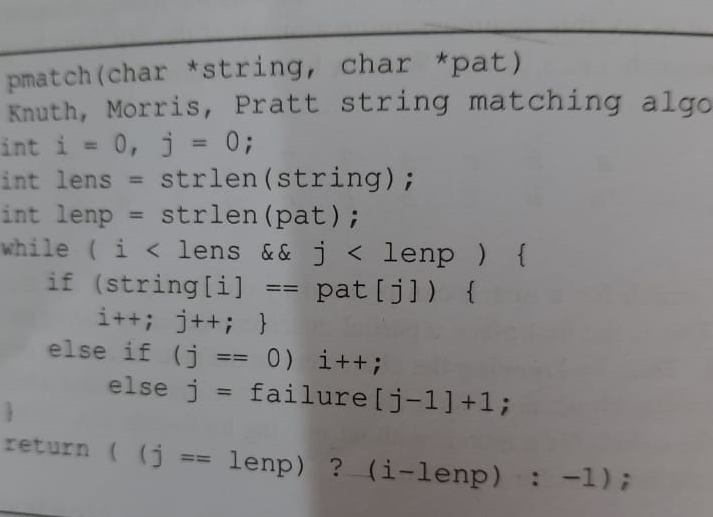
failure[j]=i+1;

else

failure[j]=-1

}

}



abaabaab

-1-1001234

Ababaa