$_tstrlen(constchar*$
$str);$
$_tstrlen(constchar*$
$str)constchar*s;for(s=str;*s;++s)return(s-str);$
$O(m*$
$n)$§**??**
$_advance=$
$haystack;for(p2=$
$needle[1];*p2;++$
$p2)p1_advance++;//advancep1_advanceM-1times$
$_advance;p1_advance+$
$+)char*p1_old=(char*)p1;p2=needle;while(*p1*p2*p1==*p2)p1++;p2++;if(!*p2)returnp1_old;$
$_old+$
$1;returnNULL;$
$_2 8$
$_MAX(2147483647)orINT_MIN(-2147483648)isreturned.$
$_MAX/10||(num==$
$INT_MAX/10(str[i]-'$
$0')>$
$INT_MAXreturnsign==$
$-1?INT_MIN:$
$INT_MAX;num=$
$num*$
$10+$
$str[i]-'$
$0';returnnum*$
$sign;$
$_8$
NM
$O(MN)$
$_1$

$_prefix(constchar*$
$pattern,intnext[])inti;intj=-1;constintm=strlen(pattern);$
$_prefix(pattern,next);$
$_2$

$_string_search_algorithm.htmlBoyer-$
$Moorealgorithm,http://www-igm.univ-mlv.fr/lecroq/string/node14.html$
$_3$

$_moore.c]/*$
$**$
$http:$
$//www-$
$igm.univ-$
$mlv.fr/lecroq/string/node14.html*$
$*$
$suffixes(),pre_gs()*$
$/include<$
$stdio.h>$
$include<$
$stdlib.h>$
$include<$
$string.h>$
$_right(constchar*$
$pattern,intright[])inti;constintm=strlen(pattern);$
$_gs(constcharpattern[],intgs[])inti,j;constintm=strlen(pattern);int*suff=(int*)malloc(sizeof(int)*(m+1));$
$_moore(constchar*$
$text,constchar*$
$pattern)inti,j;intright[ASIZE];/*bad-charactershift*/constintn=strlen(text);constintm=strlen(pattern);int$
$_right(pattern,right);pre_gs(pattern,gs);$
$_moore(text,pattern);printf("return0;$
$_4$

$_karp.c]include<$
$stdio.h>$
$include<$
$string.h>$
$^(M-$
$1)*$
$@returni+$
$1M*$
$/staticlongrehash(constlongh,constcharfirst,constcharnext,constlongRM)longnewh=(h+Q-RM*firstnewh=$
$_karp(constchar*$
$text,constchar*$
$pattern)inti;constintn=strlen(text);constintm=strlen(pattern);constlongpattern_hash=hash(pattern,m);longtex$

Robert
Sedgewick-
http://book.douban.com/subject/10432347/
Robert
Sedgewick-
http://book.douban.com/subject/10432347/
BOYER
R.S.,
MOORE