1] Brute Force Matching

#include <iostream>

#include <conio.h>

#include <string.h>

using namespace std;

int bruteForceMatch(char\* T, char\* P)

{

int lenT = strlen(T);

int lenP = strlen(P);

int i, j;

for (i = 0; i <= (lenT - lenP); i++)

{

j = 0;

while (j < lenP && T[i + j] == P[j])

{

j += 1;

if (j == lenP)

return i;

}

}

return 0;

}

int main()

{

char \*T = new char[20];

char \*P = new char[20];

cout << "Brute Force Matching\n";

cout << "enter string:";

cin >> T;

cout << "Enter pattern:";

cin >> P;

int index = bruteForceMatch(T, P);

if (index)

cout << "\nPattern found at index " << index;

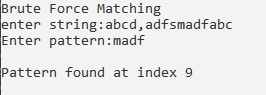
else

cout << "\nPattern not present in string";

\_getch();

return 0;

}



2] Boyer- Moore Matching

#include <iostream>

#include <conio.h>

#include <string.h>

using namespace std;

int last(char\* P, char c, int len)

{

for (int i = len - 1; i >= 0; i--)

if (c == P[i])

return i;

return -1;

}

int min(int a, int b)

{

return a < b ? a : b;

}

int bmMatch(char\* T, char\* P) {

int lenT = strlen(T);

int lenP = strlen(P);

int i, j;

i = j = lenP - 1;

do

{

if (P[j] == T[i])

{

if (j == 0)

return i; //match

else

{

i -= 1; j -= 1;

}

}

else

{

i += (lenP - min(j, 1 + last(P, T[i], lenP)));

}

} while (i < lenT - 1);

return 0;

}

int main()

{

char \*T = new char[20];

char \*P = new char[20];

cout << "Boyer Moore Matching\n";

cout << "enter string:";

cin >> T;

cout << "Enter pattern:";

cin >> P;

int index = bmMatch(T, P);

if (index)

cout << "\nPattern found at index " << index;

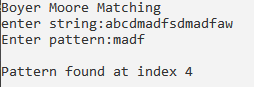
else

cout << "\nPattern not present in string";

\_getch();

return 0;

}



3)KNUTH MORRIS –PRATT ALGORITHM

#include <stdio.h>

#include<conio.h>

#include <string.h>

#include <ctype.h>

int main()

{

char string[100], matchcase[20], c;

int i = 0, j = 0, index;

/\*Scanning string\*/

printf("Enter string: ");

do

{

fflush(stdin);

c = getchar();

string[i++] = tolower(c);

} while (c != '\n');

string[i - 1] = '\0';

/\*Scanning substring\*/

printf("Enter substring: ");

i = 0;

do

{

fflush(stdin);

c = getchar();

matchcase[i++] = tolower(c);

} while (c != '\n');

matchcase[i - 1] = '\0';

for (i = 0; i < strlen(string) - strlen(matchcase) + 1; i++)

{

index = i;

if (string[i] == matchcase[j])

{

do

{

i++;

j++;

} while (j != strlen(matchcase) && string[i] == matchcase[j]);

if (j == strlen(matchcase))

{

printf("Match found from position %d to %d.\n", index + 1, i);

\_getch();

return 0;

}

else

{

i = index + 1;

j = 0;

}

}

}

printf("No substring match found in the string.\n");

\_getch();

return 0;

}

