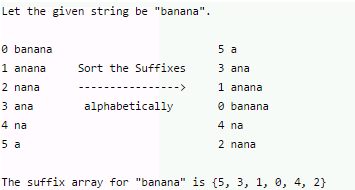
**NLOGN ALGORITHM**

ALGORITHM:



* Assign a rank to all suffixes using ASCII value of first character.
* For every character, we also store rank of next adjacent character, i.e., the rank of character at str[i + 1] . If a character is last character, we store next rank as -1
* Sort all Suffixes according to rank and adjacent rank. Rank is considered as first digit or MSD, and adjacent rank is considered as second digit.
* **Sort according to first four character**  
  Assign new ranks to all suffixes. To assign new ranks, we consider the sorted suffixes one by one. Assign 0 as new rank to first suffix. For assigning ranks to remaining suffixes, we consider rank pair of suffix just before the current suffix. If previous rank pair of a suffix is same as previous rank of suffix just before it, then assign it same rank. Otherwise assign rank of previous suffix plus one.
* For every suffix str[i], also store rank of next suffix at str[i + 2]. If there is no next suffix at i + 2, we store next rank as -1
* Sort all Suffixes according to rank and next rank.

**CODE**

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

char \*substring(char\*,int,int);

int main()

{

int length,index1,index2,index3,rank[10],next\_rank[10],pass,hold,new\_rank[10],index[10],count=1,new\_nextrank[10],new\_count=1;

char string1[10]="banana";

char substring1[10][10],substring2[10][10],substring3[10],\*pointer,substring4[10][10];

char \*ptr;

int position=3,length1,hold1;

ptr=string1;

//Initializing the arrays to zero

for(index1=0;index1<10;index1++)

{

rank[index1]=0;

next\_rank[index1]=0;

new\_rank[index1]=0;

new\_nextrank[index1]=0;

}

printf("\n Input string = ");

while(\*ptr!='\0')

{

printf("%c",\*ptr);

ptr++;

}

printf("\n");

length = strlen(string1);

printf("\nlength=%d\n",length);

for(index1=0;index1<length;index1++)

{

index[index1]=index1;

}

printf("----------step-1----------\n");

for(index1=0;index1<length;index1++)

{

printf("index[%d]=%d\n",index1,index[index1]);

}

for(index1=0;index1<length;index1++)

{

for(index2=0;string1[index2]!='\0';index2++)

{

substring1[index1][index2] = string1[index1+index2];

}

substring1[index1][index2] = '\0';

}

printf("\n");

printf("The suffixs are\n");

for(index1=0;index1<length;index1++)

{

for(index2=0;substring1[index1][index2]!='\0';index2++)

{

printf("%c",substring1[index1][index2]);

}

printf("\n");

}

printf("\n");

for(index1=0;index1<length;index1++)

{

rank[index1]=substring1[index1][0]-'a';

next\_rank[index1]=substring1[index1][1]-'a';

}

for(index1=0;index1<length;index1++)

{

printf("rank[%d]=%d\n",index1,rank[index1]);

}

for(index1=0;index1<length;index1++)

{

printf("next\_rank[%d]=%d\n",index1,next\_rank[index1]);

}

printf("---------step-2------------\n");

for(index1=0;index1<length;index1++)

{

for(index2=index1+1;index2<length;index2++)

{

if(rank[index1]>rank[index2])

{

hold=rank[index1];

rank[index1]=rank[index2];

rank[index2]=hold;

hold=next\_rank[index1];

next\_rank[index1]=next\_rank[index2];

next\_rank[index2]=hold;

hold=index[index1];

index[index1]=index[index2];

index[index2]=hold;

}

}

}

for(index1=0;index1<length;index1++)

{

printf("rank[%d]=%d\n",index1,rank[index1]);

}

for(index1=0;index1<length;index1++)

{

printf("next\_rank[%d]=%d\n",index1,next\_rank[index1]);

}

for(index1=0;index1<length;index1++)

{

printf("index[%d]=%d\n",index1,index[index1]);

}

for(index1=0;index1<length;index1++)

{

if(rank[index1]==rank[index1+1])

{

count++;

}

else

{

printf("count=%d\n",count);

for(index2=0;index2<count;index2++)

{

for(index3=index2+1;index3<count;index3++)

{

if(next\_rank[index2]>next\_rank[index3])

{

hold=next\_rank[index2];

next\_rank[index2]=next\_rank[index3];

next\_rank[index3]=hold;

hold=index[index1];

index[index1]=index[index2];

index[index2]=hold;

}

}

}

count=1;

}

}

for(index1=0;index1<length;index1++)

{

printf("next\_rank[%d]=%d\n",index1,next\_rank[index1]);

}

for(index1=0;index1<length;index1++)

{

for(index2=index1+1;index2<length;index2++)

{

if(rank[index1]==rank[index2]&&next\_rank[index1]==next\_rank[index2])

{

hold=index[index1];

index[index1]=index[index2];

index[index2]=hold;

}

}

}

for(index1=0;index1<length;index1++)

{

printf("index[%d]=%d\n",index1,index[index1]);

}

printf("\n----------step-3------------\n");

for(index1=1;index1<length;index1++)

{

if(rank[index1]==rank[index1-1]&&next\_rank[index1]==next\_rank[index1-1])

{

new\_rank[index1]=new\_rank[index1-1];

}

else

new\_rank[index1]=new\_rank[index1-1]+1;

}

for(index1=0;index1<length;index1++)

{

printf("new\_rank[%d]=%d\n",index1,new\_rank[index1]);

}

for(index2=0;index2<length;index2++)

{

strcpy(substring2[index2],substring1[index[index2]]);

}

printf("\n");

for(index1=0;index1<length;index1++)

{

printf("%s",substring2[index1]);

printf("\n");

}

for(index1=0;index1<length;index1++)

{

strcpy(substring4[index1],substring2[index1]);

}

for(index1=0;index1<length;index1++)

{

length1=strlen(substring2[index1]);

if (length1<3)

{

new\_nextrank[index1]=-1;

}

else

{

pointer=substring(substring2[index1],position,length1);

printf("Req substring=%s\n",pointer);

for(index2=0;index2<length;index2++)

{

if(strcmp(pointer,substring4[index2])==0)

{

new\_nextrank[index1]=new\_rank[index2];

}

}

free(pointer);

}

}

for(index1=0;index1<length;index1++)

{

printf("new\_nextrank[%d]=%d\n",index1,new\_nextrank[index1]);

}

for(index1=0;index1<length;index1++)

{

if(new\_rank[index1]==new\_rank[index1+1])

{

count++;

hold=new\_nextrank[index1];

new\_nextrank[index1]=new\_nextrank[index1+1];

new\_nextrank[index1+1]=hold;

hold=index[index1];

index[index1]=index[index1+1];

index[index1+1]=hold;

}

else

{

printf("count=%d\n",count);

count=1;

}

}

for(index1=0;index1<length;index1++)

{

printf("new\_nextrank[%d]=%d\n",index1,new\_nextrank[index1]);

}

for(index1=0;index1<length;index1++)

{

printf("index[%d]=%d\n",index1,index[index1]);

}

for(index2=0;index2<length;index2++)

{

strcpy(substring2[index2],substring1[index[index2]]);

}

printf("\n");

for(index1=0;index1<length;index1++)

{

printf("%s",substring2[index1]);

printf("\n");

}

return 0;

}

char \*substring(char \*substring2,int position,int length1)

{

char \*pointer;

int c;

pointer=malloc(length1+1);

if(pointer==NULL)

{

printf("Unable to allocate memory\n");

exit(EXIT\_FAILURE);

}

for(c=0;c<position-1;c++)

substring2++;

for(c=0;c<length1;c++)

{

\*(pointer+c)=\*substring2;

substring2++;

}

\*(pointer+c)='\0';

return pointer;

}

