# Algorithm Code Book

Tanvir Hasan Anick

July 8, 2015

# Contents

# Chapter 1

# Data Structure

# Chapter 2

# Graph Theory

# Chapter 3

# Flow networks/ matching

# Chapter 4

# Dynamic programming

# Chapter 5

# Strings

## 5.1 KMP

Tutorial

```
#include<bits/stdc++.h>
using namespace std;
char TXT[10000000],ptr[10000000];
vector<int> compute_prefix(const char *p)
{
    int m=strlen(p+1);
    vector<int> prefix(m+1);
    prefix[1]=0;
    int k=0;
    for(int i=2; i<=m; i++)
    {
        while(k>0 and p[k+1]!=p[i])k=prefix[k];
        if(p[k+1]==p[i])k=k+1;
        prefix[i]=k;
    }
    return prefix;
}
vector<int> KMP_match(const char *txt,const char *ptrn)
{
    int n=strlen(txt+1);
    int m=strlen(ptrn+1);
    vector<int> Prefix=compute_prefix(ptrn);
    vector<int>Match_position;
    int q=0;
    for(int i=1; i<=n; i++)
    {
        while(q>0 and ptrn[q+1]!=txt[i])q=Prefix[q];
        if(ptrn[q+1]==txt[i])q=q+1;
        if(q==m)
        {
            Match_position.push_back(i-m);
            q=Prefix[q];
        }
    }
```

```
35      return Match_position;
36  }
37  int main()
38  {
39      scanf("%s %s",TXT+1,ptr+1);
40      vector<int> Match_position=KMP_match(TXT,ptr);
41      for(int i=0; i<Match_position.size(); i++)
42      {
43          if(!i)printf("%d",Match_position[i]);
44          else printf(" %d",Match_position[i]);
45      }
46      return 0;
47  }
```

# Chapter 6

# Computational geometry

# Chapter 7

# Math

# Chapter 8

# Number Theory