### Suffix Array + LCP, O(nlogn)

### Tested on UVa 11512 GATACCA, LightOJ 1347 - Aladdin and the Magical Lamp

### struct entry {

### int fst,snd, p;

### bool operator <(const entry &b) const

### {

### if(fst==b.fst) return snd<b.snd;

### else return fst<b.fst;

### }

### } L[MX];

### string s, tmp;

### int P[20][MX],s1, s2, sl,stp;

### void suffix\_array() //from pdf

### {

### for (int i = 0; i < sl; i ++)

### P[0][i]=s[i]-'A';

### stp=1;

### for (int cnt = 1; cnt >> 1 < sl; stp ++, cnt <<= 1)

### {

### for (int i = 0; i < sl; i ++)

### {

### L[i].fst = P[stp - 1][i];

### L[i].snd = i + cnt < sl ? P[stp - 1][i + cnt] : -1;

### L[i].p = i;

### }

### sort(L, L + sl);

### for (int i = 0; i < sl; i ++)

### P[stp][L[i].p] = i > 0 && L[i].fst == L[i - 1].fst && L[i].snd == L[i - 1].snd ? P[stp][L[i - 1].p] : i;

### }

### }

### int lcp(int x, int y)

### {

### int k, ret = 0;

### if (x == y) return sl - x;

### for (k = stp-1; k >= 0 && x < sl && y < sl; k --)

### if (P[k][x] == P[k][y])

### x += 1 << k, y += 1 << k, ret += 1 << k;

### return ret;

### }