

Soruda [L, R] araligi yerine [1, R] hesaplayip [1, L - 1]'i cikarabiliriz. Ilk R eleman icin bunu yapmak icinse olusacak sekli bir agac gibi dusunebiliriz. Sonra bu agactaki ilk R leaf'i toplamaliyiz. Agacin derinligi en fazla $\log(10^{18})$ olacaktir cunku her seferinde 2'yle genisler her node en az. Bu sayede tree, sparse table, gibi bir structure ile bu agaci toplayabiliriz.

```
#include <bits/stdc++.h>
using namespace std;
```

```
typedef long long ll;
typedef pair<int, int> ii;
```

```
const int N=1e5+555,M=70;
```

```
int n, q;
ll m, e;
char s[N];
string rep[11];
ll go[M][4][4], len[M][4];
ll sum[N], sumg[N][4];
ll A, B, C, D;
```

```
void f(ll x) {
    if(!x) return;
    ll ee = e;
    ll a = 0;
    int l=0,r=n;
    while(l<r){
        int m=(l+r+1)>>1;
        if(sum[m]<=x)
            l=m;
        else
            r=m-1;
    }
    x-=sum[l];
    A += sumg[l][0];
    B += sumg[l][1];
    C += sumg[l][2];
    D += sumg[l][3];
    if(x){
        int c=s[l+1];
        while(x and e){
            e--;
            for(auto ch:rep[c]){
                if(len[e][ch]<=x){
                    x-=len[e][ch];
                    A += go[e][ch][0];
                }
            }
        }
    }
}
```

```

        B += go[e][ch][1];
        C += go[e][ch][2];
        D += go[e][ch][3];
    }
    else {
        c=ch;
        break;
    }
}
}
}
e = ee;
}

```

```

int to[333];

```

```

int main() {
    to['A'] = 0;
    to['C'] = 1;
    to['G'] = 2;
    to['T'] = 3;
    cin >> n >> m;
    for(int i = 0; i < 4; i++) {
        cin >> rep[i];
        for(auto& c : rep[i])
            c = to[c];
        go[0][i][i] = 1;
        len[0][i] = 1;
    }
    scanf("%s", s + 1);
    for(int i = 1; i <= n; i++)
        s[i] = to[s[i]];
    for(int i = 1; i < M; i++) {
        for(int j = 0; j < 4; j++) {
            for(auto c : rep[j]) {
                for(int k = 0; k < 4; k++)
                    go[i][j][k] += go[i - 1][c][k];
                len[i][j] += len[i - 1][c];
            }
        }
    }
    ll cur = n;
    while(cur < m) {
        e++;
        cur = 0;
        for(int i = 1; i <= n; i++)

```

```

    cur += len[e][s[i]];
}
// printf("cur = %lld\n", cur);
for(int i = 1; i <= n; i++) {
    sum[i] = sum[i - 1] + len[e][s[i]];
    for(int j = 0; j < 4; j++)
        sumg[i][j] = sumg[i - 1][j] + go[e][s[i]][j];
}
cin >> q;
while(q--){
    ll x, y;
    scanf("%lld %lld", &x, &y);
    A = B = C = D = 0;
    f(x - 1);
    A *= -1;
    B *= -1;
    C *= -1;
    D *= -1;
    f(y);
    printf("%lld %lld %lld %lld\n", A, B, C, D);
}
return 0;
}

```