# USACO DEC07 Problem 'bclgold' Analysis

**by Christos Tzamos**

This problem can be solved with dynamic programming on the intervals of cows but there is also a simple greedy strategy.

Between the two cows in the edges, you must always pick the cow with the smallest initial letter. If both cows have the same initial letter in order to decide you must look a little bit deeper and check the second cows in the line's edges or the third ones if those are equal and so on until you find two cows that are different. Then you pick the cow from the side of the smallest one.

This process can be summarized as follows.

At any given interval [a,b] with string S([a,b]) you choose:

Cow a if S([a,b]) < rev( S([a,b]) )
Cow b otherwise
where rev(S) is the reverse string e.g. rev("abc") = "cba"

This can be implemented in O(N^2) but we can achieve O(NlogN) by using suffix arrays.

Here are the two implementations:

The O(N^2)
```
#include<cstdio>

char S[2010],ln=0;

void prnt(char a) {
        if(ln==80) {printf("\n");ln=0;}
        printf("%c",a);ln++;
}

int main() {
        int i,j,N,pi,pj,val;
        freopen("bcl.in" ,"r",stdin );
        freopen("bcl.out","w",stdout);
        scanf("%d",&N);
        for(i=0;i<N;i++) scanf(" %c ",S+i);
        i=0,j=N-1;
        while(i<=j) {
                if(S[i]<S[j])           {prnt(S[i]);i++;}
                else if(S[i]>S[j])      {prnt(S[j]);j--;}
                else {
                        pi=i+1;pj=j-1;val=S[i];
                        while( pj-pi>1 && S[pi]==S[pj]) {pi++,pj--;}
                        if(S[pi]<S[pj]) prnt(S[i]),i++;
                        else prnt(S[j]),j--;
                }
        }
        printf("\n");
        return 0;
}
```

## And the O(NlogN)

```
#include<cstdio>
#include<cstring>
#include<cstdlib>

#define MAXN 500050

char S[2*MAXN];
int N,ln=0;
int o[2][2*MAXN], t[2*MAXN][2];
int A[2*MAXN], B[2*MAXN], C[2*MAXN], D[2*MAXN];

void prnt(char a) {
        if(ln==80) {printf("\n");ln=0;}
        printf("%c",a);ln++;
}

int main() {

        int i, j, jj, x, k;

        freopen("bcl.in" ,"r",stdin );
        freopen("bcl.out","w",stdout);
        scanf("%d",&N);
        for(i=0;i<N;i++) {
                scanf(" %c ",S+i);
                S[N+i] = S[i];
        }

        memset(A, 0, sizeof(A));
        for (i = 0; i < 2*N; ++i) A[(int)(S[i]-'A')] = 1;
        for (i = 1; i < 26; ++i) A[i] += A[i-1];
        for (i = 0; i < 2*N; ++i) o[0][i] = A[(int)(S[i]-'A')];
        x=0;
        for (j = 0, jj = 1, k = 0; jj < N && k < 2*N; ++j, jj <<= 1) {

                memset(A, 0, sizeof(A));
                memset(B, 0, sizeof(B));

                for (i = 0; i < N; ++i) {
                        ++A[ t[i][0] = o[x][i] ];
                        ++B[ t[i][1] = (i+jj<N) ? o[x][i+jj] : 0 ];
                }

                for (i = N; i < 2*N; ++i) {
                        ++A[ t[i][0] = o[x][i] ];
                        ++B[ t[i][1] = (i-jj>=N) ? o[x][i-jj] : 0 ];
                }

                for (i = 1; i <= 2*N; ++i) {
                        A[i] += A[i-1];
                        B[i] += B[i-1];
                }

                for (i = 2*N-1; i >= 0; --i)
                        C[--B[t[i][1]]] = i;

                for (i = 2*N-1; i >= 0; --i)
                        D[--A[t[C[i]][0]]] = C[i];

                x ^= 1;
                o[x][D[0]] = k = 1;
                for (i = 1; i < 2*N; ++i)
                        o[x][D[i]] = (k += (t[D[i]][0] != t[D[i-1]][0] ||
                                t[D[i]][1] != t[D[i-1]][1]));
        }
        i=0,j=N-1;
        while(i<=j) {
                if(S[i]<S[j])           {prnt(S[i]);i++;}
                else if(S[i]>S[j])      {prnt(S[j]);j--;}
                else if(o[x][i]<o[x][N+j]) {prnt(S[i]);i++;}
                else {prnt(S[j]);j--;}
        }
        printf("\n");
}
```