

TOI推廣計畫

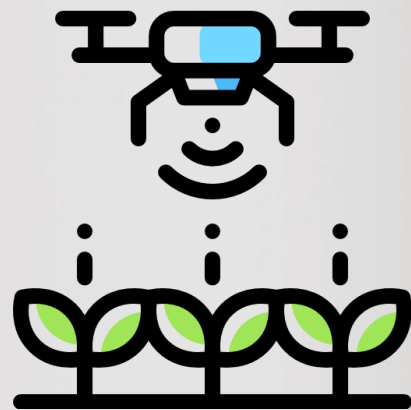
解題－農田灌溉



題目

一位農夫引進了最新型的灌溉裝置，此種灌溉裝置可藉由調整電力強度來增減農田灌溉的範圍，電力分五個等級，而電力越強，則範圍越大，且耗費的電量較高。若電力強度為 N ，則可灌溉以自身為中心，向左右延伸 $(N - 1)$ 單位的範圍(如電力為2，則可灌溉最多3單位的範圍)，今天農夫想要以最低成本的電費來灌溉農田，請你寫一個程式幫他計算，最低成本的電費以及電力強度為多少。

每台灌溉裝置的電費
計算的公式為： $1.32 \times N + 1.07$ 。



Icon made by <https://www.flaticon.com/authors/freepik> from www.flaticon.com

輸入格式

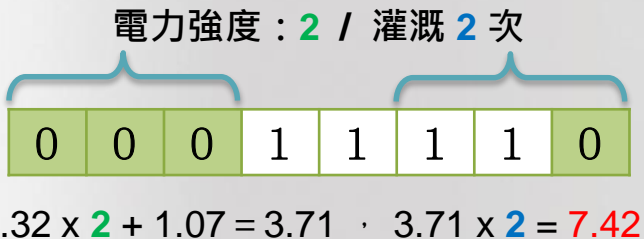
每筆測試資料為二列，

第一列有一個正整數 L ($1 \leq L \leq 5,000,000$)，代表道路的總長度。

第二列共有 L 個字元，字元0代表有農田的道路，字元1代表無種植作物的道路，灌溉裝置可以放在道路上任何一個地方，且灌溉裝置的電力強度必須統一。

輸出格式

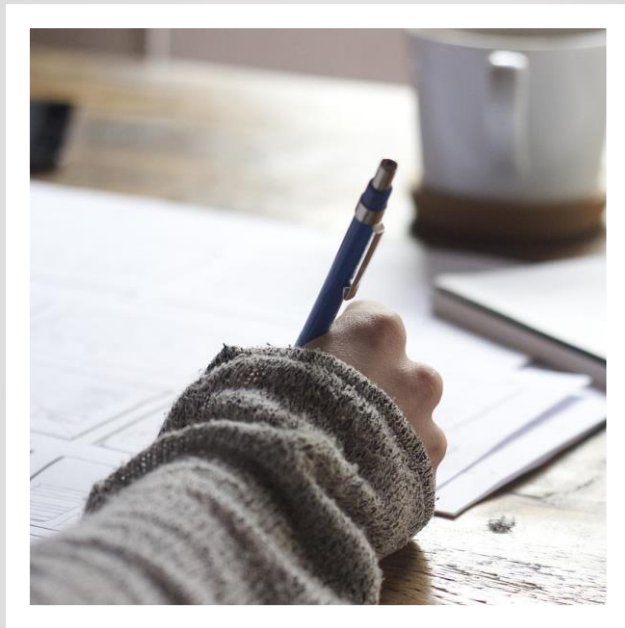
對每筆資料請輸出一列共兩個數字，請輸出灌溉所有農田的最低成本以及電力強度，成本輸出至小數點後兩位，兩個數字間以一個空白隔開，若成本相同，則輸出電力強度較大者。



輸入範例	輸出範例
8 00011110	7.42 2

解題重點:

1. Greedy
2. 搜尋最佳解



◆ 重點一：Greedy

由陣列第0項開始遍歷，當看到0時，代表其附近需有灌溉裝置，選擇將灌溉裝置放在最大距離的位置，便可灌溉最多農田，以電力等級3為例，農田如下圖：

i	0	1	2	3	4	5	6	7
farm[i]	0	0	0	1	1	1	1	0

i	0	1	2	3	4	5	6	7
farm[i]			*					*

◆ 重點一：Greedy

i	0	1	2	3	4	5	6	7
farm[i]	0	0	0	1	1	1	1	0

i	0	1	2	3	4	5	6	7
farm[i]			*					*

因farm[0]為農田，因此選擇將灌溉裝置放在**最大距離(0+2)的位置**上，便可使farm[0~4]皆被灌溉，接著從farm[5]繼續往下遍歷尋找下一個農田的位置，找到其位於farm[7]。

◆ 重點二：搜尋最佳解

將電力等級1~5皆計算過一次，並計算電費最低者即可。

i	0	1	2	3	4	5	6	7
farm[i]	0	0	0	1	1	1	1	0

等級一

i	0	1	2	3	4	5	6	7
farm[i]	*	*	*					*

等級二

i	0	1	2	3	4	5	6	7
farm[i]		*						*

◆ 重點二：搜尋最佳解

將電力等級1~5皆計算過一次，並計算電費最低者即可。

等級三

i	0	1	2	3	4	5	6	7
farm[i]				*				


等級四

i	0	1	2	3	4	5	6	7
farm[i]					*			

等級五

i	0	1	2	3	4	5	6	7
farm[i]						*		

◆ 參考解答

```
#include<bits/stdc++.h>
const int sizes=10000005;
char farm[sizes]={};
int main()
{
    int L;
    scanf("%d",&L);
    for(int i=0;i<L;i+=1)
    {
        scanf(" %c",&farm[i]);
    }
    
    printf("%.2f %d\n",large,ind);
}
```

```
double large=99999999;
int ind=-1;
for(int i=1;i<=5;i+=1)
{
    int range=i*2-1;
    int num=0;
    for(int j=0;j<L;j+=range)
    {
        while(farm[j]=='1' && j<L)
        {
            j++;
        }
        num+=(j<L);
    }
    double money=(1.32*i+1.07)*num;
    if(large>=money)
    {
        large=money;
        ind=i;
    }
}
```