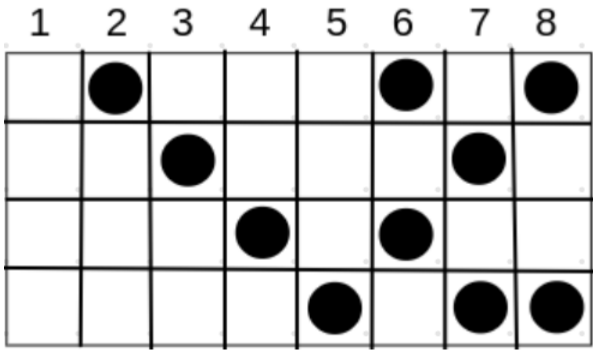
1 Wprowadzenie

Na szachownicy z 4 x n polami, do których jest przypisane są liczby naturalne chcemy poukładać kamienie tak, żeby zmaksymalizować sumę liczb na zajętych polach. Nie chcemy, żeby dwa kamienie ze sobą sąsiadowały (pola ze wspólną krawędzią) oraz na jednym polu może być tylko jeden kamyk.

Łatwo zauważyć, że mamy 8 możliwości ustawienia kamyków w jednej kolumnie.



Rysunek 1: Wszystkie możliwości

Możliwość ustawienia kamyków w jednej kolumnie jest zależne od sąsiednich kolumn.

2 Algorytm

Nasz algorytm będzie dynamikiem, gdzie problem sprowadzamy do szachownicy o mniejszej liczbie kolumn i mając wcześniej obliczone maksymalne wartości jakie możemy uzyskać dla k-tej kolumny (dla każdej możliwości ułożenia ostatniej kolumny) to jesteśmy w stanie obliczyć wyniki dla k+1 kolumny.

```
long long chessboard(int *tab, int n){
    long long dynam[n+1][8];
    for(int g = 0; g < 8; g++)
        dynam[0][g] = 0;
    for(int g = 0; g <= n; g++){
        for(int h = 0; h < 8; h++){
            dynam[g][h] = value(tab, g, h) + maxprev(dynam, g, h);
        }
    }
    long long ret = 0;
    for(int g = 0; g < 8; g++)
        ret = max(ret, dynam[n][g]);
    return ret;
}
```

Rysunek 2: Pseudokod algorytmu

Funkcja value oblicza wartość ułożenia kamyków w g-tej kolumnie dla h-tego sposobu, zaś funkcja maxprev zwraca maksymalną wartość jaką mogła mieć g-1 kolumna, która może sąsiadować z h-tym sposobem ułożenia kamyków. Pod koniec zwracamy maksimum z obliczonych wartości dla każdej możliwości ustawienia ostatniej kolumny.

Szukana wartość jest maksimum z sposobów ułożenia dla ostatniej kolumny.

3 Dowód poprawności

Dowód będzie indukcyjny względem k-tej kolumny.

Teza indukcji: dla k-tej kolumny mamy obliczone największe wartości, które można otrzymać dla każdego sposobu ułożenia kolumny.

Baza indukcji:

k = 1 wtedy obliczamy tylko wartość dla każdej kolumny, ponieważ maxprev zwróci zero.

Krok indukcyjny:

Założenie indukcyjne: teza jest spełniona dla k-tej kolumny Teza indukcyjna: teza jest spełniona dla k+1. kolumny

Wiemy, że maksymalna wartość dla k+1. kolumny i h-tego sposobu, jest to suma wartości kamyków z tej kolumny oraz maksymalna wartość jaką mogliśmy otrzymać wcześniej, czyli maksymalna wartość dla k-tej kolumny z ułożeniem, które nie koliduje z naszym, a ponieważ z założenia indukcyjnego mamy już te możliwości dla k obliczone to możemy je obliczyć też dla k+1.

4 Złożoność

Wyliczenie wartości dla każdej kolumny jest wykonane w czasie stałym (mamy ograniczoną liczbe możliwość i ograniczoną liczbe możliwych sąsiadów), a ponieważ musimy wykonać obliczenia dla każdej kolumny to złożoność naszego algorytmu wynosi $\Theta(n)$.