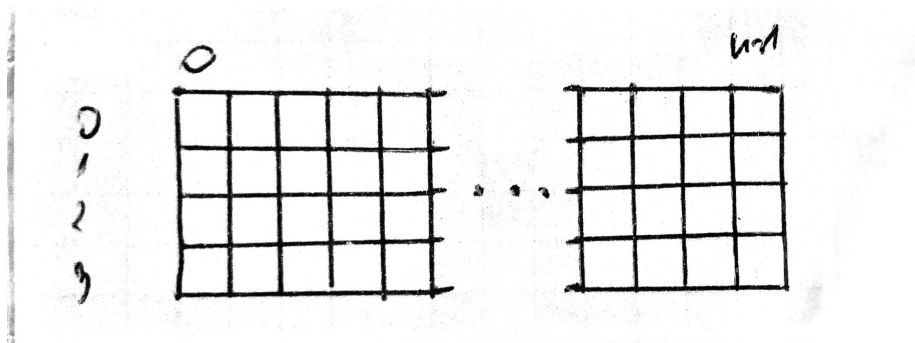


Karol Cidyło

Zadanie 8 z listy 4.

Mamy szachownice o rozmiarze 4 x n.



Rysunek 1: Szachownica 4 x n.

Zadaniem jest rozmieszczenia na szachownicy kamyków tak, żeby na każdym polu znajdował się co najwyżej jeden kamień oraz pola szachownicy, na których są kamienie nie mogą mieć wspólnych krawędzi. Uznajmy, że na szachownicy jest kamień jeśli pole ma wartość 1, w przeciwnym przypadku 0. Możemy zauważyć, że jeżeli jeden wymiar szachownicy to 4, wtedy możliwe rozmieszczenia kamieni można wyrazić za pomocą następujących masek bitowych:

```
0 0 0 0
0 0 0 1
0 0 1 0
0 1 0 0
0 1 0 1
1 0 0 0
1 0 0 1
1 0 1 0
```

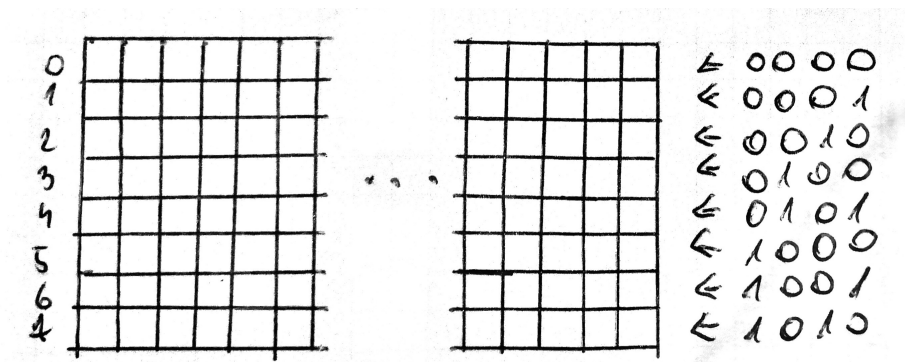
Mamy ich konkretnie 8.

Algorytm

Rozwiązaniem jest stworzenie szachownicy 8 x n (Rysunek 2), gdzie $s[i][j]$ będzie mówić jaka jest maksymalna wartość szachownicy utworzonej przez pierwsze i kolumn przy ustawieniu j kolumny i . Czyli mając wymiar 8 zawieramy każdą możliwą maske.

$$s[i][j] = \begin{cases} 0 & \text{gdy } i < 0 \\ \max_{k=\text{poprawne}(j)}(s[i-1][k] + \text{zysk}(i, k)) & \text{w.p.p.} \end{cases}$$

Funkcja **poprawne(j)** zwraca możliwe maski (ustawienia kamieni) tak, aby były zachowane wytyczne zadania. Dla masek jest to po prostu sytuacja gdzie usta-



Rysunek 2: Szachownica 8 x n.

wiając dwie jedna pod drugą w żadnej kolumnie z wyrażenia \mathbf{a} & \mathbf{b} nie otrzymujemy 1, gdzie \mathbf{a} jest wartością z wiersza pierwszego, \mathbf{b} z drugiego. Algorytm ten można zaimplementować jako automat, mamy skończoną i znaną liczbę masek zgodnych między sobą. Drugim sposobem jest implementacja zgodnych masek jako listy dla każdej z 8 dostępnych. Można zauważyć, że tylko dla maski zerowej (0 0 0 0) mamy 7 możliwości, dla innych będzie mniej.

$\text{zysk}(\mathbf{i}, \mathbf{k})$ jest to zysk jaki otrzymamy z ustawienia kamieni w kolumnie \mathbf{i} -tej przy masce \mathbf{k} . To rozwiązujemy za pomocą sumy mnożeń kolejnych wartości maski z wartościami w kolumnie \mathbf{i} -tej.

Wynikiem jest $\max_{k=0..7} s[n][k]$.

Złożoność obliczeniowa

Wynik mamy w $s[n][k]$ dla $k=0..7$, więc musimy przejść całą tablicę 4 x n. Dla każdego elementu ustawienie zgodnych masek możemy ustalić w czasie stałym. Dla każdego pola musimy też wybrać maks z jednego z 8 pól, jednak będzie to średnio mniej, ponieważ nie wszystkie są ze sobą zgodne. Także złożoność tego algorytmu to $O(n)$.

Złożoność pamięciowa.

Tablica 8 x n oraz ewentualna pamięć na pamiętanie wartości w aktualnie sprawdzanej kolumnie oraz listy zgodnych masek. $O(n)$.