



oczywiście, taki indeks operacji, przed wykonaniem której bilans na koncie jest możliwie *najmniejszy!*

```
function max_fragment3(var a : array of LongInt;
                      n : LongInt) : LongInt;
var i, j, wyn, suma, min_bilans : LongInt;
    bilans : array[0 .. MAX_N] of LongInt;
begin
    bilans[0] := a[0];
    for i := 1 to n do
        bilans[i] := bilans[i - 1] + a[i];
    wyn := 0;
    min_bilans := 0;
    for j := 1 to n do begin
        suma := bilans[j] - min_bilans;
        if suma > wyn then
            wyn := suma;
        if bilans[j] < min_bilans then
            min_bilans := bilans[j];
        end;
    max_fragment3 := wyn;
end;
```



Programy i testy opisane w artykule są dostępne na stronie [deltami.edu.pl](http://deltami.edu.pl)

Tym razem główna pętla wykonuje jedynie kilka prostych operacji, więc standardowo z pominięciem stałej określamy złożoność czasową ostatecznego rozwiązania jako  $O(n)$ . I rzeczywiście, jak można się już domyślić, to rozwiązanie bez problemu radzi sobie z wszystkimi trzema testami.

W tym artykule spróbowałam w kilku słowach opowiedzieć, na czym polega analiza złożoności czasowej algorytmów. Celowo pominęliśmy kwestie takie jak dobór odpowiednich typów danych (czy typ `LongInt` wystarcza?) oraz *złożoność pamięciową*, czyli – znów przybliżone – określenie zużycia pamięci przez program.

## Hexapawn, czyli czego można nauczyć pudełka

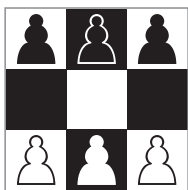
Kamila ŁYCZEK

Maszyną górnolotnie będzie nazywany zestaw pudełek opatrzonych w etykiety, koraliki oraz stosowną instrukcję obsługi.

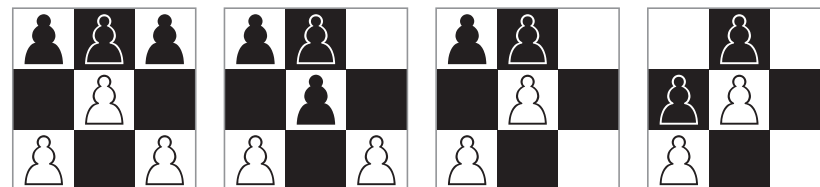
Uczenie się oznacza adaptacyjne zmiany w systemie, w tym sensie, że w miarę postępu procesu te same zadania wykonywane są przynajmniej tak samo dobrze jak we wcześniejszych etapach.

Zamiast analizować, czy gra jest sprawiedliwa, zamiast szukać najlepszych strategii graczy, można stworzyć pewną maszynę, która część tej pracy wykona za nas. Trzeba jej objaśnić zasady, a potem z nią grać, niekoniecznie najlepiej – w końcu jeszcze nie przeanalizowaliśmy gry. Maszyna, grając, zapamiętując i wyciągając wnioski z przegranych oraz wygranych (co śmiało można zakwalifikować jako *uczenie się*), prędzej czy później zorientuje się, jak grać możliwie najlepiej, a więc ogrywać nas, o ile to tylko możliwe.

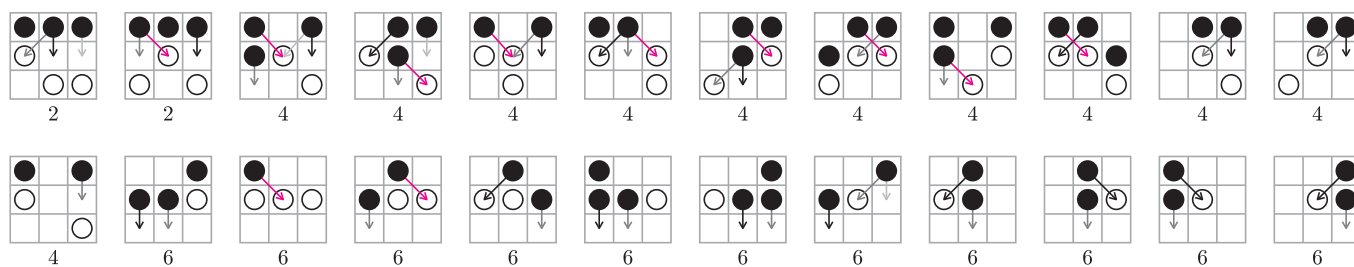
**Instrukcja gry Hexapawn dla ludzkich graczy.** Gra rozgrywa się na szachownicy  $3 \times 3$ . Początkowe ustawienie przedstawia rysunek 1. Dwaj gracze (pierwszy – biały, drugi – czarny) ruszają się na przemian. W każdym ruchu gracz rusza się jednym ze swoich pionków jedno pole do przodu lub, jeżeli ma taką możliwość, może (nie musi) wykonać bicie pionka przeciwnika jedno pole po skosie do przodu. Wygrywa ten gracz, który jako pierwszy dotrze swoim pionkiem na przeciwną stronę szachownicy (na jedno z pól, z których grę rozpoczął przeciwnik) lub który uniemożliwi jakikolwiek ruch przeciwnikowi (zbije wszystkie jego pionki bądź go zablokuje).



Rys. 1. Początkowe ustawienie w grze Hexapawn.



Rys. 2. Przykładowa rozgrywka zakończona wygraną gracza czarnego.



Rys. 3. Etykiety maszyny. Numer pod każdą etykietką oznacza, w którym z kolei ruchu w grze dana sytuacja może zaistnieć.

Faktycznie istnieją więcej niż 24 różne sytuacje w rozgrywce. W drugim ruchu mogą powstać trzy, a nie dwie możliwości (inaczej niż na rysunku 3). Jednak ruch białego gracza w pierwszym ruchu skrajnym pionkiem, niezależnie czy prawym czy lewym, prowadzi do analogicznych sytuacji. Rozważenie tylko jednej z nich przyspiesza uczenie.

**Instrukcja gry dla maszyny.** Droga Maszyno, w każdym swoim ruchu (jesteś graczem drugim, czarny kolor pionków) możesz znaleźć się w jednej z sytuacji przedstawionych na rysunku 3. Masz wtedy do dyspozycji tylko te ruchy, które są oznaczone strzałkami. Wybierasz jeden z dostępnych Ci ruchów. Następnie rusza się gracz biały. Po nim ponownie wykonujesz Twój ruch, znowu wybierasz strzałkę, aż... znajdziesz się w sytuacji, której nie ma na rysunku, oznacza to koniec gry. Powodzenia, Maszyno!

**Instrukcja techniczna gry (dla nauczyciela maszyny).** Maszyna składa się z 24 pudełek, każde opatrzone etykietą (rys. 3) przedstawiającą sytuację, która może zaistnieć w grze oraz możliwe do wykonania w tej sytuacji ruchy (kolorowe strzałki). Do każdego pudełka należy włożyć tyle koralików w kolorach odpowiadających strzałkom. W każdym pudełku znajdzie się tyle koralików, ile strzałek jest na jego etykietce, w kolorach takich jak strzałki. W parzystych ruchach należy odszukać pudełko z odpowiadającą grze sytuacją, a następnie losowo wykonać ruch (wylosować koralik i wykonać takie posunięcie, jakie przedstawia strzałka w jego kolorze). Koralik należy zostawić przed pudełkiem aż do końca partii.

Jeżeli grę wygra maszyna, wkładamy wszystkie wyjęte podczas partii koraliki z powrotem do odpowiednich pudełek. Jeżeli grę wygra gracz biały, usuwamy z ostatniego użytego pudełka wylosowany koralik (odpowiadający ostatniemu w grze ruchowi maszyny). Rozpoczynamy kolejną grę z jednym koralikiem mniej lub z taką samą liczbą koralików. Usunięcie koralika z ostatniego pudełka będzie skutkowało tym, że kiedy maszyna ponownie znajdzie się w analogicznej sytuacji, nie wykona ruchu, który wcześniej doprowadził ją do przegranej (pamięta porażki).

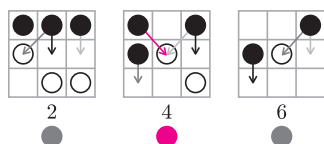
Jeżeli pudełko zostanie puste, usuwamy je z maszyny. Jeżeli maszyna znajdzie się w takiej sytuacji, oznacza to, że nie może wykonać żadnego ruchu, należy traktować to jako jej przegraną (mimo że według reguł dla ludzkich graczy istnieją dozwolone ruchy).

Opisaną metodą będą sukcesywnie eliminowane wszystkie ruchy prowadzące do przegranej drugiego gracza. Pozostaną tylko posunięcia nieprowadzące do przegranej. Jak łatwo się przekonać, maszyna staje się mądrzejsza tylko po przegranej partii. Po dokładnie 11 przegranych (co oznacza około 40 gier z myślącym przeciwnikiem) maszyna nauczy się ogrywać nas zupełnie (w grze *Hexapawn* strategię wygrywającą ma drugi gracz). Nauka będzie tym szybsza, im lepszy przeciwnik maszyny.

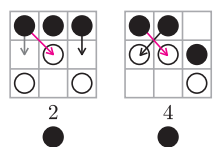
**Nieco szerzej.** Gra *Hexapawn* została wymyślona przez Martina Gardnera w celu zilustrowania prostego działania *maszyny samouczącej się*, co opisał w rubryce *Mathematical Games* w *Scientific American* w marcu 1962. Maszynę nazwał *HER* (ang. *Hexapawn Educable Robot*) i stworzył dla niej partnera, samouczącego się robota *HIM* (ang. *Hexapawn Instructable Matchboxes*), który był graczem wykonującym pierwszy ruch (i wszystkie nieparzyste) w ich wspólnych partiach.

Maszyna może być zaprojektowana w nieco inny sposób. Jeżeli zależy nam nie na tym, aby jak najszybciej stała się mistrzem, lecz na tym, żeby np. przegrała jak najmniejszą liczbę partii podczas 20 rozgrywek, wtedy warto nie tylko karać (wyjmując koraliki), ale nagradzać za wygraną poprzez dokładanie koralików w odpowiednich kolorach do wszystkich pudełek użytych w wygranej rozgrywce.

Analogiczne maszyny mogą zostać zbudowane do rozgrywania innych gier. Pierwsza tego typu maszyna rozgrywała grę kółko i krzyżyk, składała się z 300 pudełek. Została opisana w 1961 roku przez Donalda Michie, biologa z Uniwersytetu w Edynburgu.



Rys. 4. Przykładowa rozgrywka wygrana przez maszynę. Żaden koralik nie jest usuwany.



Rys. 5. Po wykonaniu czwartego ruchu, oznaczonego czarną strzałką, biały gracz wygra. Jego jedynym możliwym ruchem będzie poruszenie się pionkiem znajdującym się na środku planszy jedno pole do przodu. Maszyna przegra. Z powyższego pudełka z numerem 4 usuwamy czarny koralik.

Artykuł Gardnera o maszynie samouczącej jest przedrukowany m.in. w książce *Knots and Borromean Rings, Rep-Tiles, and Eight Queens: Martin Gardner's Unexpected Hanging*, The New Martin Gardner Mathematical Library, 2014.