

# Sprawozdanie-Algorytm Mini Max z odcięciem alfa-beta

---

## Dane autora

Marcin Polewski  
331 425

---

## Cel zadania

Celem zadania było zaimplementowanie algorytmu mini max z odcięciem alfa-beta, które ogranicza konieczność przeszukiwania pod drzewa gry, które przy optymalnej grze obu graczy były nieosiągalne. Następnie należało zbadać wpływ głębokości przeszukiwań na jakość rezultatów

---

## Specyfika problemu

### **Przegrywa ten co zabierze ostatnią zapałkę**

Dla gry nim definiuje się pojęcie nimsuma – czyli dla danego kopca zamieniamy liczbę elementów na system binarny, a potem robimy XOR.

Jeśli ten nim sum na początku jest różny od zera, to drugi gracz ma wygrywającą pozycję pod warunkiem że gra optymalnie. A jeśli nim sum na początku jest równy zero, to pierwszy gracz ma wygrywającą pozycję

---

## Analiza gry - specyficzne sytuacje na bazie wiedzy eksperckiej

### **Jedna kupka z dowolną liczbą elementów**

- gracz którego to jest ruch jest na wygrywającej pozycji – wystarczy że zabierze wszystkie zapałki oprócz jednej. Jest to bardzo dobra pozycja

### **Wszystkie kupki mają po jednym elemencie**

- Jeśli liczba kupek jest nieparzysta, to aktualny gracz jest na przegranej pozycji
- jeśli liczba kupek jest parzysta, aktualny gracz jest na wygrywającej pozycji

**W przeciwnych przypadkach oceniamy pozycje na podstawie nimsuma**

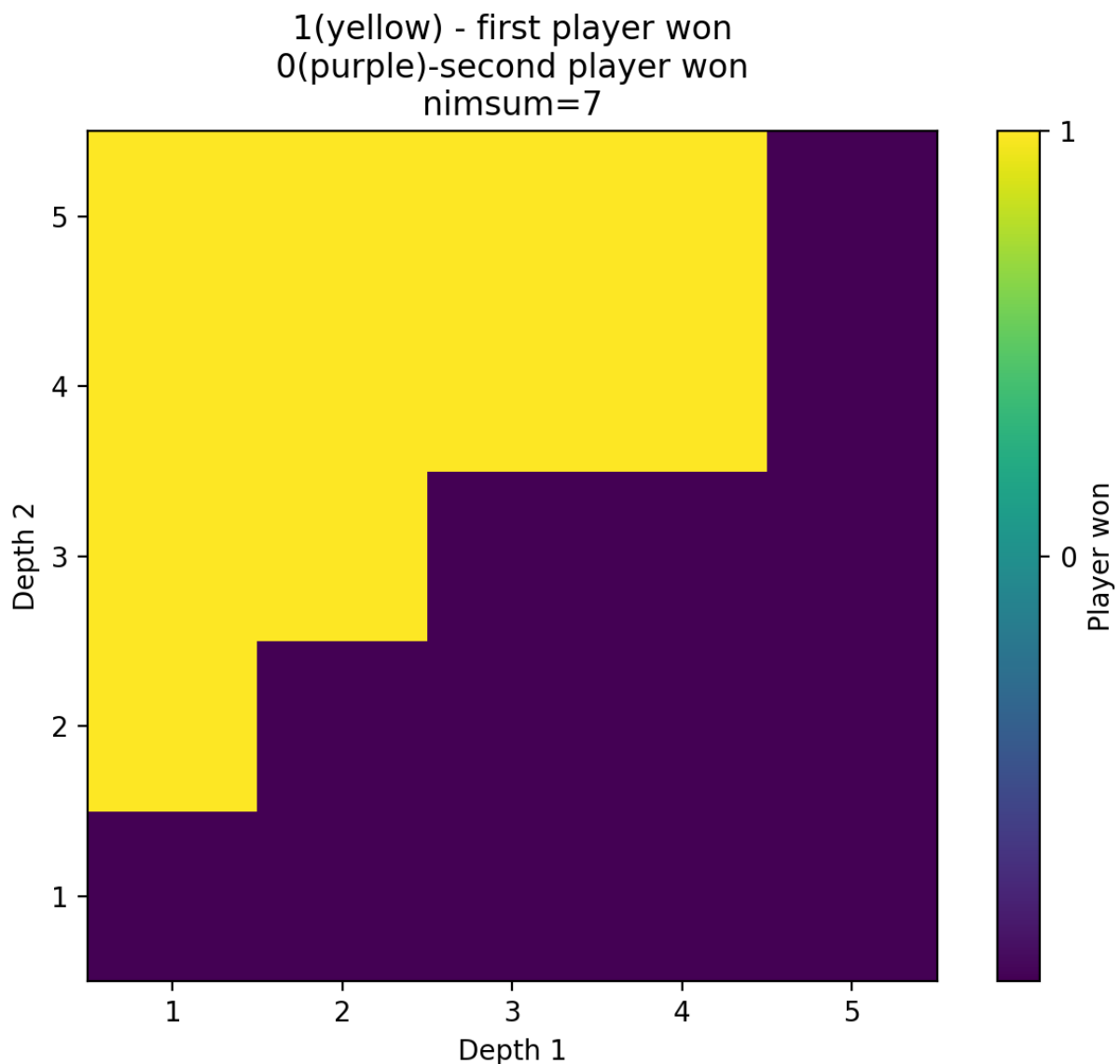
- Nimsum == 0 <- W takiej sytuacji obecny gracz jest na przegranej pozycji
- Nimsum != 0 <- W takiej sytuacji obecny gracz jest na wygranej pozycji

(stworzono na podstawie [link](#))

## Badania dla wyboru pierwszego, a nie losowego ruchu z równie dobrych

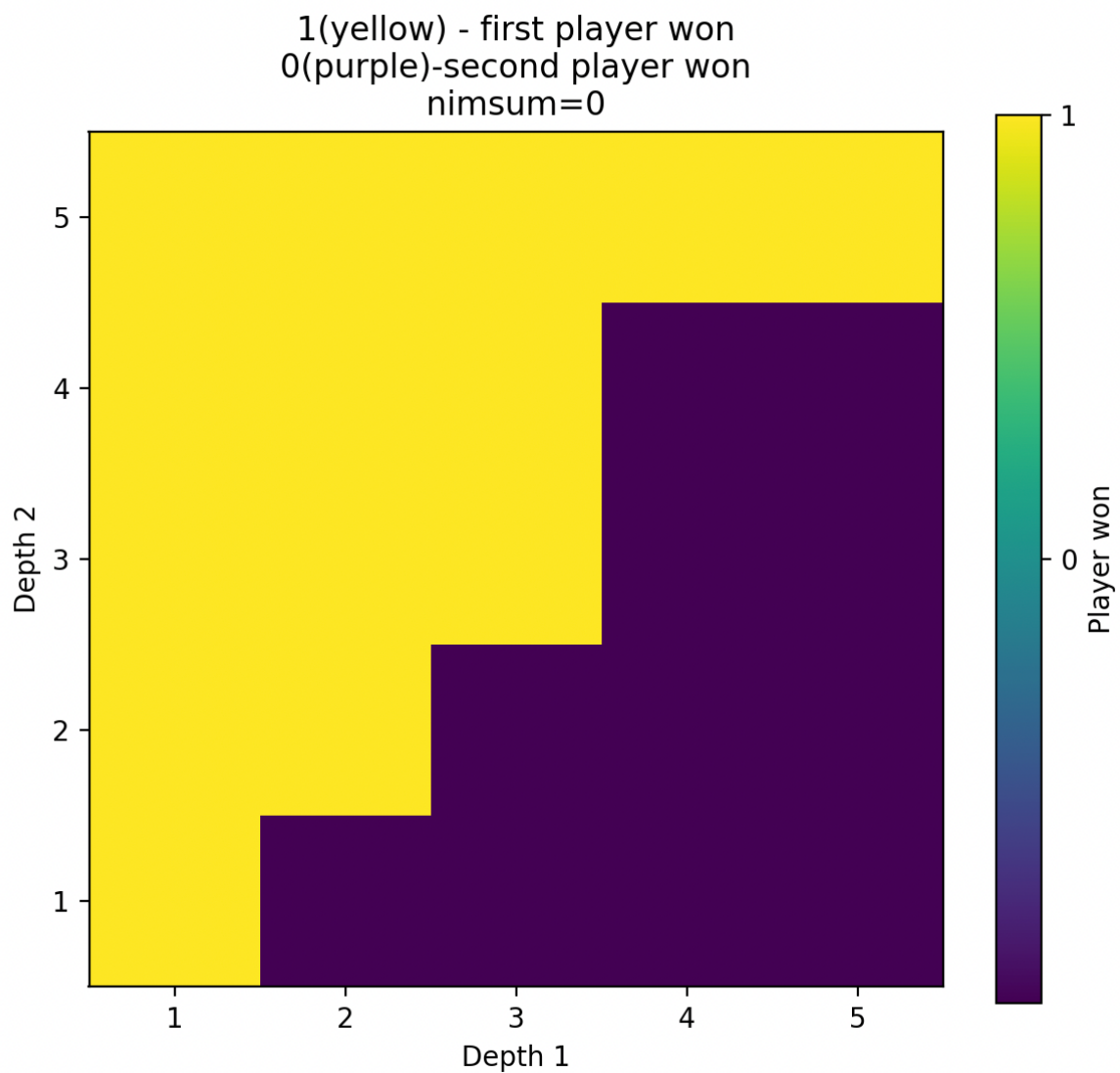
Co ciekawe, w tym przypadku algorytmy dają lepsze efekty w przypadku, gdy bierze pierwszy, najlepszy ruch, a nie losowy. Dzieje się tak moim zdaniem, gdyż funkcja zwracająca możliwe ruchy na początku zwraca te, w których wyciągamy mniej elementów z kupki, co za pewne jest pożądanym zachowaniem

Eksperyment dla nimsum!=0, dla domyślnej planszy(7,7,7)



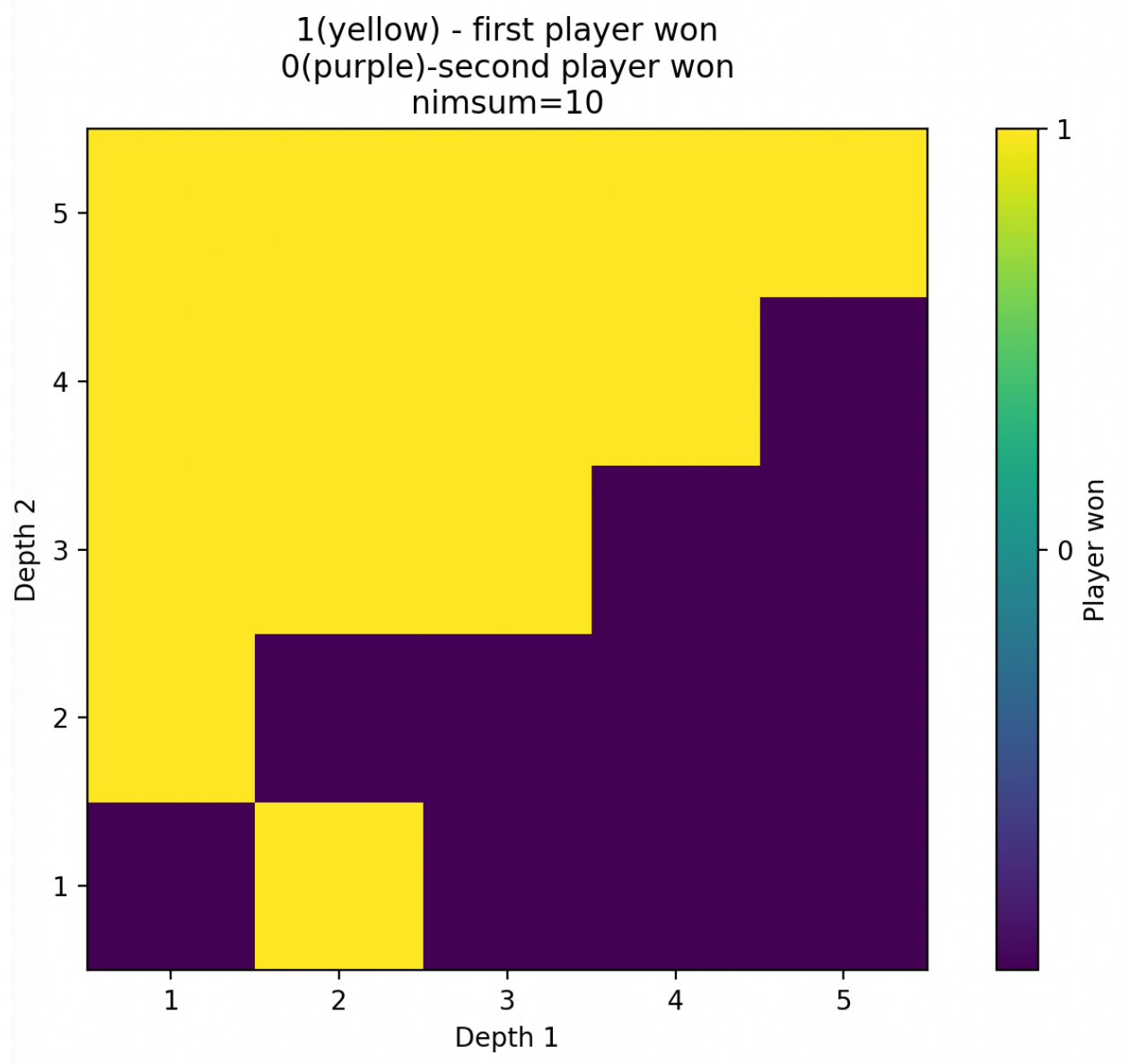
Na wykresie widoczna jest zależność między głębokością przeszukiwania, a wygraną – im większa głębokość przeszukiwania, tym większa szansa na wygraną. Widoczna jest lekka przewaga gracza drugiego, która prawdopodobnie wynika ze specyfiki gry – to znaczy zaczyna on z wygranej pozycji i jeśli gra optymalnie to wygra

Eksperyment dla nimsum=0, dla planszy (1,3,5,7)



Na wykresie jest widoczna analogiczna zależność między głębokością przeszukiwania, a wynikiem. Tym razem jednak gracz pierwszy zaczyna z korzystniejszej pozycji, co odzwierciedla jego większa liczba zwycięstw niż w poprzednim wypadku

Eksperyment dla nimsum!=0, dla domyślnej planszy(10,10,10)



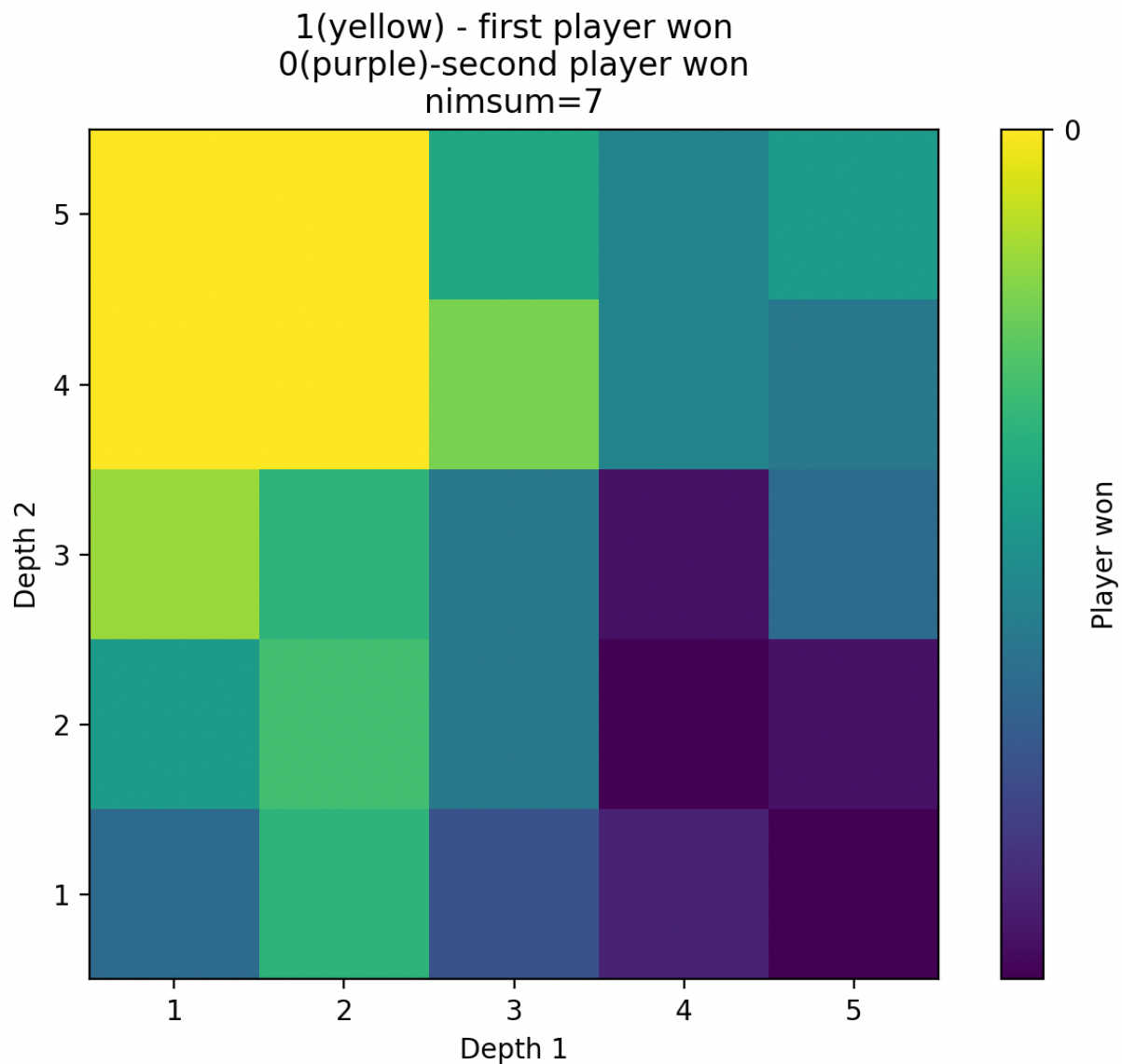
Ponownie, zależność zostaje zachowana, z drobnymi zakłóceniami.

---

## Badania dla wyboru losowego ruchu z równie dobrych

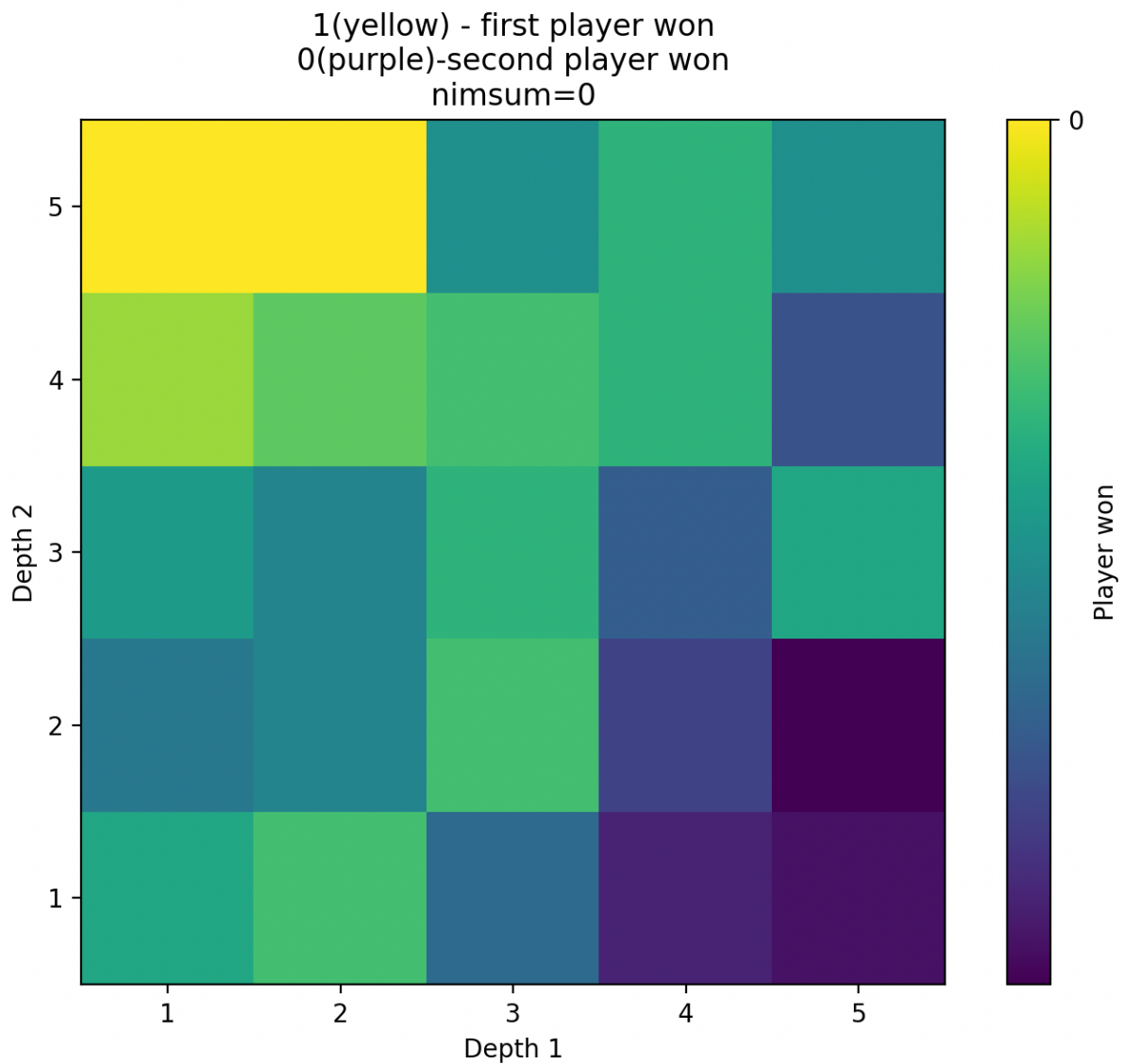
W tym przypadku wykonano każdy eksperyment 10 razy, a następnie wyniki uśredniono, a następnie wyniki uśredniono

Eksperyment dla  $\text{nimsum} \neq 0$ , dla domyślnej planszy (7,7,7)



Nie jest to wielkie zaskoczenie, że zależność wciąż zostaje zachowana. Dodatkowo widzimy, że dla równych wartości głębokości przeszukiwań, każdy z graczy wygrywa mniej więcej tyle samo razy

Eksperyment dla  $\text{nimsum} == 0$ , dla planszy (1,3,5,7)



Wynik tego przypadku jest analogiczny do poprzedniego

---

### Podsumowanie:

- Zaobserwowano, że większa głębokość przeszukiwań prowadzi do lepszych rezultatów
- W grze nim wychodzi na to, że opłaca się raczej wykonywać ruchy, w których zabieramy jak najmniej elementów