

Dany jest n -elementowy wektor liczb całkowitych z przedziału $[-10;10]$. Dwóch agentów gra w grę turową polegającą na tym, że w swojej turze agent może zabrać z wektora skrajną liczbę, tj. tę z początku wektora, bądź z końca. Gra kończy się, gdy wszystkie liczby zostaną zabrane z wektora. Celem każdego agenta jest wybrać liczby tak, aby suma zebranych przez niego liczb była jak największa. Przykładowo dla wektora $[2, 0, 3, 1]$ najlepsze możliwe rozegranie gry dla gracza rozpoczynającego to $[2, 3]$, a gracz drugi ma wtedy $[0, 1]$, lub $[1, 0]$.

Do ćwiczenia dostarczone są dwa pliki:

- `main.py`, zawierające implementacje agentów grające w grę i funkcje pomocnicze
- `test.py`, zawierający testy jednostkowe. Istniejących testów nie wolno zmieniać, ich nieuzasadniona edycja będzie skutkowała wyzerowaniem liczby punktów. Można natomiast dopisać swoje testy.

Polecenie:

Proszę dokończyć implementację agenta `MinMaxAgent` wykorzystującego algorytm `MiniMax`. Powinien on przechodzić wszystkie testy jednostkowe. Następnie proszę przetestować go w starciu z agentem posługującym się heurystyką zachłanną (`GreedyAgent`) w 1000 rozgrywkach dla $n=15$ i następujących głębokości drzewa stanów (1, 2, 3, 15). Należy w tabelce raportować średni czas wykonania się gry, średnią i odchylenie standardowe sum punktów uzyskanych w rozgrywkach. Jak głębokość drzewa wpływa na wyniki uzyskiwane przez `MinMaxAgent`-a?

Następnie proszę dla głębokości drzewa 2 oraz 15 potraktować sumę punktów uzyskiwanych przez `MinMaxAgent`-a w każdej rozgrywce jako zmienną losową. Jaki jest jej rozkład? Dobrym sposobem na wizualizację rozkładu jest stworzenie histogramu.

Proszę powtórzyć powyższe testy zamieniając `GreedyAgent`-a na `NinjaAgent`-a, oraz `GreedyAgent` na drugiego `MinMaxAgent`a (głębokość 15). Jak na podstawie histogramu stwierdzić, który agent jest lepszy? Jak przyspieszyć działanie `MinMaxAgent`a? Jak poprawić jego działanie dla drzewa przeszukiwań o głębokości 1? Uzyskane tabelki, wnioski i 6 podpisanych histogramów proszę zamieścić w raporcie w formacie pdf.

Raport oraz pliki proszę spakować do pliku o nazwie `WSI-3-NAZWISKO-IMIE.zip` i przesać na adres `grzegorz.rypesc.dokt@pw.edu.pl`.

Uwagi:

Proszę zauważyć, że aby sprawiedliwie móc porównać agenty, każdy z nich powinien rozpoczynać grę tę samą liczbą razy. W 3. linii pliku `main.py` proszę ustawić ziarno generatora liczb pseudolosowych zgodnie z komentarzem.

Za błędy językowe, nieczytelność i wklejanie zrzutów ekranu będę odejmował punkty. Szansa to nie jest prawdopodobieństwo [https://pl.wikipedia.org/wiki/Szansa_\(statystyka\)](https://pl.wikipedia.org/wiki/Szansa_(statystyka)).

Przejęcie testów nie zawsze oznacza, że algorytm działa prawidłowo.