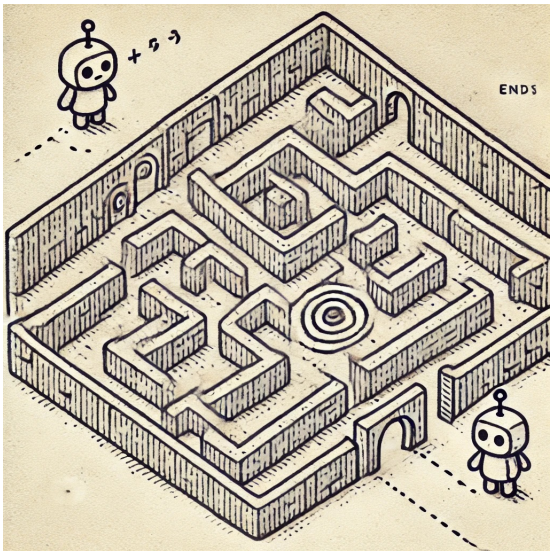
Ćwiczenie 1

Planowanie trasy robota mobilnego w siatce kwadratów pól - Algorytm A^*



Zadanie do wykonania

1) Na pulpicie utwórz folder o nazwie Imię_Nazwisko i umieść w nim wszystkie pliki związane z ćwiczeniem.

2) Przeczytaj teorię dotyczącą algorytmu A^*. Jeśli masz problemy ze zrozumieniem, przeanalizuj przykład na kartce.

3) Wygeneruj mapę z przeszkodami, używając programu map_generator.exe.

4) Zaimplementuj stosując dowolny język programowania algorytm A^* na wygenerowanej mapie. Wartość 0 oznacza dostępne miejsce, 5 to przeszkody, a 3 zaznacza wyznaczoną trasę. Start ma współrzędne (0,0) (pierwszy wiersz od dołu, pierwsza kolumna), a Cel na (19,19) (dwudziesty wiersz od dołu, dwudziesta kolumna). Użyj odległości euklidesowej jako heurystyki. Koszt pojedynczego ruchu wynosi 1. Możliwe ruchy to góra, dół, lewo i prawo (ruchy po skosie są zabronione). Kolejność przeszukiwania pól to góra, dół, lewo, prawo. Konflikty rozwiązuj hierarchicznie lub losowo. W wariancie hierarchicznym wybieraj najwcześniej lub najpóźniej napotkaną wartość spośród konfliktujących.

5) Zaimplementuj demonstrację użycia algorytmu A^* lub jego zoptymalizowanej wersji w formie graficznej, np. w formie gry. Wykorzystaj narzędzia generatywnej sztucznej inteligencji (np. ChatGPT).

Algorytm A^* - część teoretyczna

Algorytm A^* jest jedną z najefektywniejszych technik planowaniu trasy robotów mobilnych. Omówmy działanie algorytmu na siatce kwadratowych pól, zwanej gridem. Metoda A^* opiera się na zasadzie "pierwszy-najlepszy". Szuka ona ścieżki do celu o najmniejszym koszcie, mierzonej wartością funkcji f, która dla każdej rozważanej pozycji *poz* jest definiowana następująco:

$$f(poz) = g(poz) + h(poz)$$

gdzie *poz* to aktualnie rozważana pozycja, $g(poz)$ to koszt dojścia od pozycji startowej do *poz*, a $h(poz)$ to heurystyka szacująca odległość do celu z pozycji *poz*.

Heurystyka $h(poz)$ może przyjmować różne formy, ponieważ nie jest jednoznacznie określona. Ważne jest, aby była dopuszczalna, co oznacza, że nie może przeszacowywać odległości do celu. Heurystyka dopuszczalna musi spełniać następujący warunek dla każdej możliwej do odwiedzenia pozycji *poz*:

$$h(poz) \leqslant \textit{koszt optymalnej drogi do celu z pozycji poz}.$$

Jeśli ten warunek nie jest spełniony, istnieje możliwość, że w pewnych konfiguracjach pozycji startowej, pozycji docelowej oraz ustawień przeszkód, algorytm A^* wybierze nieoptymalną ścieżkę do celu.

Przejdźmy do opisania podstawowych założeń i kroków algorytmu A^* mających zastosowanie w gridzie.

Krok 1) Ustal sposób poruszania się agenta, koszt poszczególnych ruchów oraz kolejność przeszukiwania pól wokół aktualnej pozycji.

Krok 2) Wybierz heurystykę h, która spełnia warunek dopuszczalności.

Krok 3) Określ sposoby rozwiązywania konfliktów, które mogą powstać podczas eksploracji siatki.

Krok 4) Wskaż punkt startowy i cel.

Krok 5) Stwórz dwie listy pomocnicze: listę otwartą, początkowo pustą, która będzie zawierała kratki rozważane jako pola do ekspansji, oraz listę zamkniętą, inicjowaną polem startowym, do której trafią odwiedzone pola, jednocześnie usuwając je z listy otwarnej.

Krok 6) Kratki otaczające ostatnio dodane pole do listy zamkniętej, które możemy odwiedzić, trafiają na listę otwartą. Zachowują one informację o "rodzicu", czyli kratce, przez którą zostały dodane do listy otwartej. Dla każdej takiej kratki obliczana jest wartość funkcji f. Jeśli kratka już ma przypisaną wartość f z innego pola dodanego wcześniej do listy zamkniętej, porównujemy nową wartość z aktualną. Zmieniamy "rodzica" tylko wtedy, gdy nowa wartość f jest mniejsza niż poprzednia.

Krok 7) Do listy zamkniętej trafia kratka z listy otwartej o najmniejszej wartości f. Konflikty rozwiązywane są hierarchicznie — spośród kratek z tym samym oszacowaniem f wybieramy pole, które zostało odwiedzone najpóźniej.

Krok 8) Jeśli cel nie został osiągnięty i lista otwarta zawiera przynajmniej jeden element, przejdź do Kroku 6.

Krok 9) Algorytm może zakończyć się na dwa sposoby:

1. Jeśli cel został osiągnięty, wróć wstecz przez kolejne kratki rodziców do kratki startowej, wyznaczając w ten sposób optymalną ścieżkę od startu do celu.

2. Jeśli lista otwarta jest pusta, a cel nie został osiągnięty, zwróć komunikat o niemożliwości dotarcia do celu.

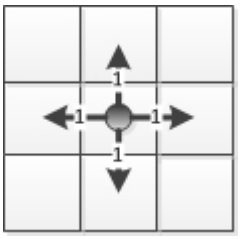
Przykład działania algorytmu A^* z ustaloną heurystyką dopuszczalną, opartą na odległości euklidesowej, przedstawiono na Rys. 2.

1. Rys. 2a pokazuje siatkę ze współrzędnymi kratek.

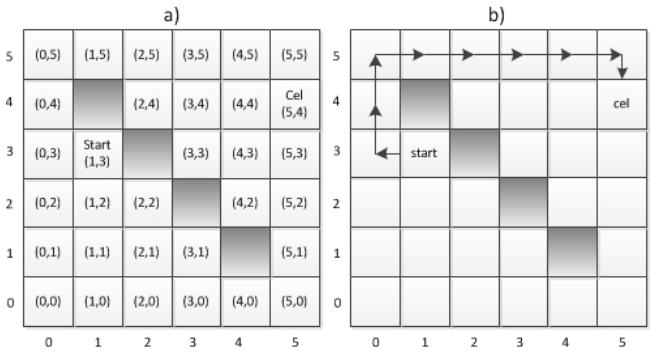
2. Rys. 2b ilustruje drogę wyznaczoną przez algorytm A^* od kratki startowej do kratki docelowej.

Dla uproszczenia zakładamy, że robot może poruszać się po pustych kratkach w czterech kierunkach: dół, lewo, góra i prawo. Robot przeszukuje kratki wokół odwiedzanego pola w tej samej kolejności i nie może poruszać się po przekątnych kratek – patrz Rys. 1.

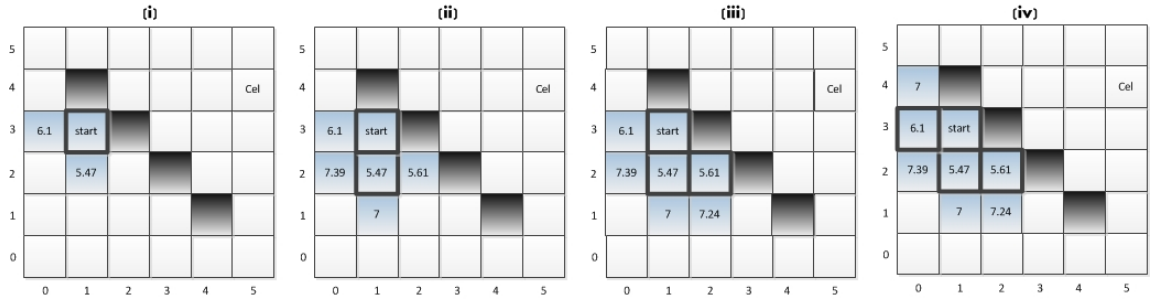
Kolejne kroki działania algorytmu A^* na siatce można prześledzić na Rys. 3, 4, 5, 6 oraz 7.



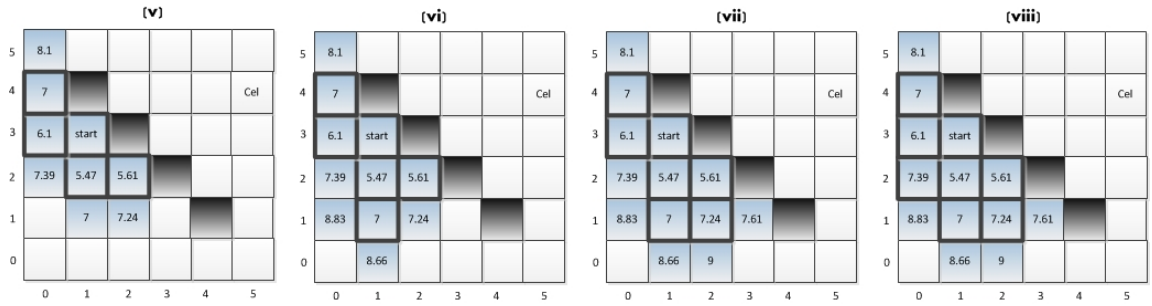
Rysunek 1: Strategia poruszania się agenta po siatce z ustalonym kosztem pojedynczego ruchu, przy założeniu wykluczenia ruchów po skosie. Kolejność przeglądania kratek wokół aktualnej pozycji: dół, lewa, góra, prawa.



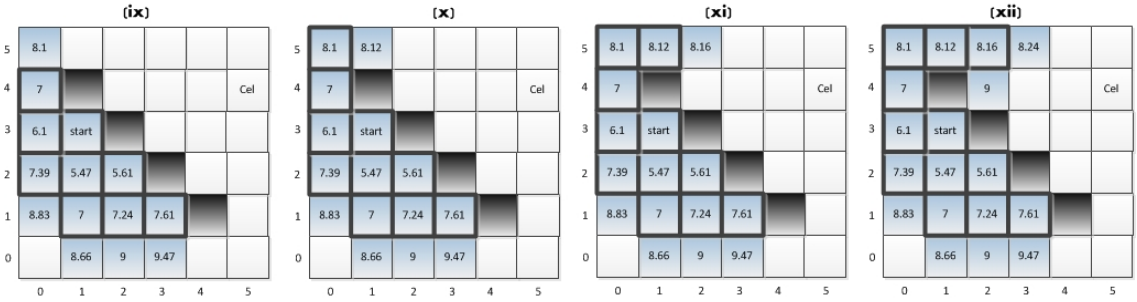
Rysunek 2: Działanie algorytmu A^* z zastosowaną dopuszczalną heurystyką opartą na metryce euklidesowej, $\mathrm{h}((\mathrm{poz}_x, poz_y)) = \sqrt{(poz_x - cel_x)^2 + (poz_y - cel_y)^2}$, gdzie (poz_x, poz_y) to współrzędne kratki, dla której wyliczamy heurystykę, a (cel_x, cel_y) to współrzędne celu. **a)** Siatka ze współrzędnymi kratek. **b)** Optymalna droga wyznaczona przez algorytm A^*. Działanie algorytmu w szczegółach przedstawiono na Rys. 3, 4, 5, 6 oraz 7.



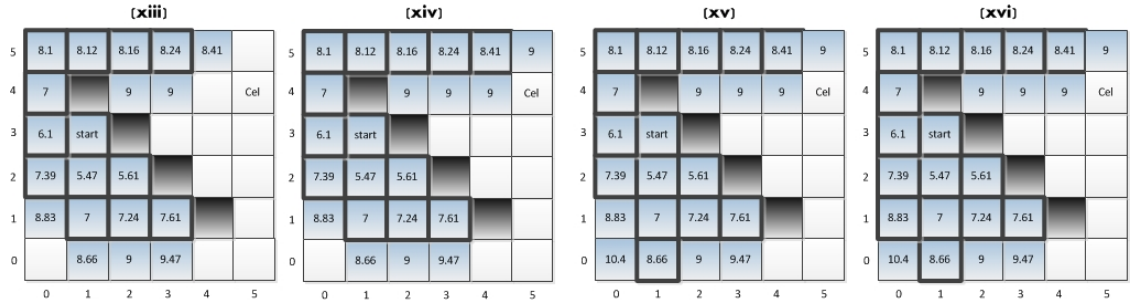
Rysunek 3: **(i)** Kratka startowa trafia na listę zamkniętą. Kratki możliwe do odwiedzenia w kolejności: dół, lewa trafiają na listę otwartą z informacją, że ich rodzicem jest kratka startowa. Obliczamy dla nich wartość funkcji f następująco: $f((1,2)) = 1 + \sqrt{(1-5)^2 + (2-4)^2} = 1 + \sqrt{20} \approx 5.47$, $f((0,3)) \approx 6.1$. W kolejnych krokach przypisujemy kratkom na liście otwartej rodziców za pośrednictwem których znalazły się na liście, z aktualizacjami rodzica, jeśli istnieje możliwość dotarcia do tych kratek w bardziej optymalny sposób. **(ii)** Wybieramy kratkę z listy otwartej, która ma najmniejszy koszt f, czyli (1,2), umieszczamy ją na liście zamkniętej, a na liście otwartej trafiają kratki przyległe. Obliczamy: $f((1,1)) = 7$, $f((0,2)) \approx 7.39$, $f((2,2)) \approx 5.61$. **(iii)** Do listy zamkniętej trafia kratka o współrzędnych (2,2). Do listy otwartej trafia kratka (2,1) z wartością $f((2,1)) \approx 7.24$. **(iv)** Do listy zamkniętej trafia kratka (0,3). Do listy otwartej trafia kratka (0,4) z wartością $f((0,4)) = 7$. Kratka (0,2) jest już na liście otwartej; wyliczona wartość f nie zmienia się przy obliczaniu z perspektywy dojścia z (0,3), stąd rodzic kratki (0,2) pozostaje bez zmian.



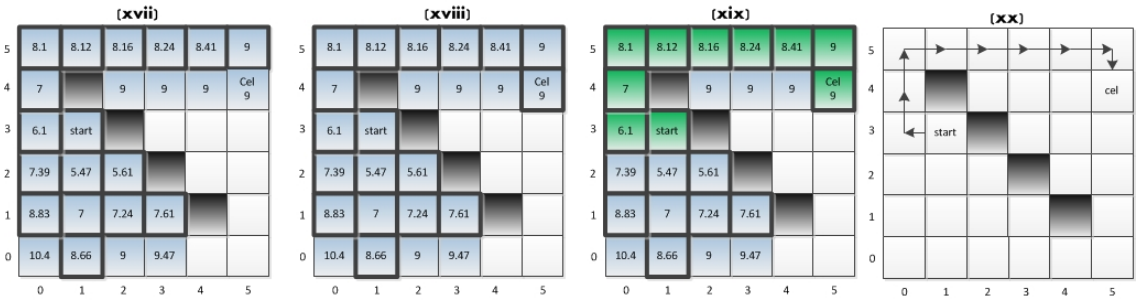
Rysunek 4: **(v)** Do listy zamkniętej kandydują dwie kratki z listy otwartej: kratka (1,1) i kratka (0,4). Korzystając z ustalonej strategii wyboru (rozwiązywania konfliktów), wybieramy kratkę, która została odkryta najpóźniej, czyli (0,4), i dodajemy ją do listy zamkniętej. Do listy otwartej trafia kratka (0,5) z wartością $f((0,5)) \approx 8.1$. **(vi)** Do listy zamkniętej trafia kratka (1,1). Do listy otwartej trafią kratki (1,0) z wartością $f((1,0)) \approx 8.66$ oraz (0,1) z wartością $f((0,1)) \approx 7.39$. Koszt kratki (2,1) nie zmienia się, stąd jej rodzicem pozostaje kratka (2,2). **(vii)** Do listy zamkniętej dodajemy kratkę (2,1). Do listy otwartej trafią kratki (2,0) z wartością $f((2,0)) \approx 9$ oraz (3,1) z wartością $f((3,1)) \approx 7.61$. **(viii)** Do listy zamkniętej trafia kratka (0,2). Stan listy otwartej nie zmienia się, ponieważ kratka (0,1) już jest na liście otwartej i jej rodzic pozostaje bez zmian.



Rysunek 5: **(ix)** Do listy zamkniętej dodajemy kratkę (3,1). Do listy otwartej trafia kratka (3,0) z wartością $f((3,0)) \approx 9.47$. **(x)** Do listy zamkniętej trafia kratka (0,5). Do listy otwartej trafia kratka (1,5) z wartością $f((1,5)) \approx 8.12$. **(xi)** Do listy zamkniętej trafia kratka (1,5). Do listy otwartej trafia kratka (2,5) z wartością $f((2,5)) \approx 8.16$. **(xii)** Do listy zamkniętej trafia kratka (2,5). Do listy otwartej trafią kratki (2,4) z wartością $f((2,4)) = 9$ oraz (3,5) z wartością $f((3,5)) \approx 8.24$.



Rysunek 6: **(xiii)** Na liście zamkniętej pojawia się kratka (3,5). Na liście otwartej trafiają kratki (3,4) z wartością $f((0,1)) = 9$ oraz (4,5) z wartością $f((4,5)) \approx 8.41$. **(xiv)** Do listy zamkniętej dopisujemy kratkę (4,5). Do listy otwartej trafiają kratki (4,4) z wartością $f((4,4)) = 9$ oraz (5,5) z wartością $f((5,5)) = 9$. **(xv)** Dodajemy do listy zamkniętej kratkę (1,0). Do listy otwartej trafia kratka (0,0) z wartością $f((0,0)) \approx 10.4$. Kratka (2,0) nie zmienia rodzica. **(xvi)** Dodajemy do listy zamkniętej kratkę (0,1). Rodzic kratki (0,0) pozostaje bez zmian.



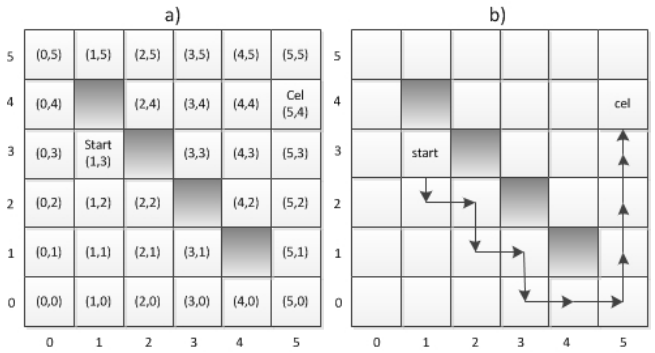
Rysunek 7: **(xvii)** Na liście otwartej znajduje się pięć kratek z minimalnym kosztem 9. Do listy zamkniętej wybieramy kratkę, która trafiła na listę otwartą jako ostatnia, czyli pole (5,5). Do listy otwartej trafia kratka cel o współrzędnych (5,4) z wartością $f((5,4)) = 9$. **(xviii)** Kratka cel trafia do listy zamkniętej. **(xix)** Algorytm przechodzi wstecz od celu do kolejnych kratek rodziców, aż do osiągnięcia kratki startowej, zapamiętując tę sekwencję. **(xx)** Pokazane są kolejne ruchy, które wyznaczają optymalną drogę od startu do celu. Algorytm kończy działanie.

Dla zobrazowania problemu, który może się pojawić, gdy heurystyka jest niedopuszczalna, przedstawiamy na Rys. 8 w podpunkcie b) sekwencję ruchów do celu na podstawie metryki rzeka, definiowanej następująco:

$$h(poz) = poz_y + |cel_x - poz_x| + cel_y$$

gdzie (poz_x, poz_y) są współrzędnymi aktualnej pozycji, a (cel_x, cel_y) są współrzędnymi celu.

Jak widzimy, przy zastosowaniu powyższej heurystyki cel został osiągnięty w sposób nieoptymalny. Obliczona ścieżka jest o dwa kroki dłuższa od optymalnej.



Rysunek 8: Działanie algorytmu A^* na bazie heurystyki określanej na podstawie metryki rzeka, ustalonej jako kratki z drugą współrzędną równą zeru. **a)** Przedstawienie siatki z wyznaczoną metryką rzeka. **b)** Sekwencja ruchów do celu na podstawie metryki rzeka.