

Wyszukiwanie i funkcje haszujące

wszelkie prawa zastrzeżone
zakaz kopiowania, publikowania i przechowywania
all rights reserved
no copying, publishing or storing

Maciej Hojda

Uwaga: Słowa „dany”, „zadany”, „podany”, „wybrany” itd. w kontekście parametrów (zmiennych) oznacza parametr zadany przez użytkownika (a nie na stałe, przez programistę), a implementacja wykorzystująca taki parametr powinna obsługiwać jego różne wartości.

Wykorzystaj strukturę z poprzedniej listy (roboty).

1 Zadanie nr 1 – wyszukiwanie liniowe

Zaimplementuj algorytm wyszukiwania robota na liście.

Wyszukiwanie odbywa się względem listy dopuszczalnych wartości parametrów robota, tzn. dla każdego parametru robota (**TYP**, **CENA**, **ZASIĘG**, **KAMERA**) zadana jest pojedyncza poszukiwana wartość, albo lista dopuszczalnych wartości, albo **None** (wartość dowolna). Algorytm zwraca parametry pierwszego znalezione robota lub „brak”, jeśli robota nie znajdzie.

Np. [„AGV”, **None**, [5, 6, 7, 8, 9, 10], 1] odpowiada wyszukiwaniu robota typu „AVG”, o dowolnej cenie, o zasięgu z przedziału 5 do 10, z kamerą.

Wejście: lista robotów (wczytywana z pliku), lista parametrów szukanego robota.

Wyjście: lista parametrów znalezione robota lub „brak”.

Uwaga: realizując zadanie zaadoptuj (być może wielokrotnie) algorytm wyszukiwania liniowego.

2 Zadanie nr 2 – wyszukiwanie binarne

Zaimplementuj kolejny algorytm wyszukiwania robota na liście.

1. Pobierz od użytkownika parametr robota (**TYP**, **CENA**, **ZASIĘG**, **KAMERA**). Posortuj listę robotów względem danego parametru. Wykorzystaj wbudowaną funkcję **sort**.
2. Zaimplementuj algorytm wyszukujący robota po liście dopuszczalnych wartości wybranego parametru. Zadanie wykonaj przy założeniu, że lista robotów jest odpowiednio posortowana. Algorytm zwraca parametry pierwszego znalezione robota lub „brak”, jeśli robota nie znajdzie.

Wejście: lista robotów (wczytywana z pliku), wybrany parametr szukanego robota i lista wartości.

Wyjście: lista parametrów znalezione robota lub „brak”.

Uwaga: realizując zadanie zaadoptuj (być może wielokrotnie) algorytm wyszukiwania binarnego.

3 Zadanie nr 3 – metoda łańcuchowa

Zaimplementuj kolejny algorytm wyszukiwania robota na liście.

1. Pobierz od użytkownika docelowy współczynnik wypełnienia tablicy α . Utwórz pustą tablicę o odpowiednim rozmiarze.
2. Wypełnij tablicę robotami rozwiązując kolizje metodą łańcuchową. Zaproponuj metodę generacji klucza.
3. Zaimplementuj algorytm wyszukujący robota po kluczu. Algorytm zwraca pierwszego znalezionego robota lub „brak”, jeśli robota nie znajdzie.

Wejście: lista robotów (wczytywana z pliku), wybrany parametr szukanego robota i wartość parametru.

Wyjście: lista parametrów znalezionego robota lub „brak”.

4 Zadanie nr 4 – adresowanie otwarte

Zaimplementuj kolejny algorytm wyszukiwania robota na liście.

1. Pobierz od użytkownika docelowy współczynnik wypełnienia tablicy α . Utwórz pustą tablicę o odpowiednim rozmiarze.
2. Zastosuj adresowanie otwarte kwadratowe do wypełnienia tablicy. Samodzielnie zaproponuj składowe funkcje haszujące i inne parametry haszowania.
3. Zaimplementuj algorytm wyszukujący robota po zadanym parametrze i zadanej jego pojedynczej wartości. Algorytm zwraca pierwszego znalezionego robota lub „brak”, jeśli robota nie znajdzie.

Wejście: lista robotów (wczytywana z pliku), wybrany parametr szukanego robota i wartość parametru.

Wyjście: lista parametrów znalezionego robota lub „brak”.

5 Zadanie nr 5 – zestawienie algorytmów

Porównaj prędkość działania algorytmów wyszukiwania zaimplementowanych w zadaniach 3 i 4.

1. W razie potrzeby, zmodyfikuj listę robotów tak, aby ich parametry CENA były unikatowe.
2. Posortuj listę robotów względem parametru CENA.
3. Wyszukuj po pojedynczej wartości parametru CENA.
4. Algorytmy wyszukiwania uruchamiaj dla kolejnych robotów, tzn. szukaj po cenach istniejących robotów (czyli zawsze zostanie znaleziony dokładnie jeden robot).
5. Wyniki przedstaw na wykresie średniego czasu wyszukiwania robota (po wszystkich robotach).
6. Wynik przedstaw dla kilku wartości współczynnika α .

Wejście: lista robotów (wczytywana z pliku), współczynnik α .

Wyjście: wykresy średniego czasu wyszukiwania robota.