

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.

1 Zadanie nr 1 – przygotowanie danych

Dalsze zadania na liście są wykonywane z wykorzystaniem struktury opisującej flotę robotów mobilnych.

1. Pojedynczy robot jest krotką o polach (parametrach):
 - (a) **TYP** – tekst ze zbioru {„AGV”, „AFV”, „ASV”, „AUV” },
 - (b) **CENA** – liczba rzeczywista z przedziału $[0, 10000]$ (PLN),
 - (c) **ZASIĘG** – liczba całkowita z przedziału $[0, 100]$ (km),
 - (d) **KAMERA** – wartość binarna $\{0, 1\}$ (jest, nie ma).
2. Utwórz strukturę danych do przechowywania listy robotów. Lista ma zadaną długość N .
3. Wypełnij strukturę robotami o parametrach generowanych losowo.
4. Zaimplementuj funkcję wyświetlającą wygenerowaną strukturę w postaci tabelki (jeden wiersz – jeden robot; równe odstępy między kolejnymi polami).
5. Zaimplementuj funkcję zapisującą/odczytującą strukturę do/z pliku.

Wejście: długość listy N .

Wyjście: lista robotów wyświetlona na ekranie i zapisana do pliku.

2 Zadanie nr 2 – 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 znajdującego 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 znajdującego robota lub „brak”.

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

3 Zadanie nr 3 – wyszukiwanie binarne

Zaimplementuj kolejny algorytm wyszukiwania robota na liście.

1. Pobierz od użytkownika parametr robota (TYP, CENA, ZASIĘG, KAMERA). Posortuj listę robotów względem zadanego 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 znalezionego robota lub „brak”, jeśli robota nie znajdzie.

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

Wyjście: lista parametrów znalezionego robota lub „brak”.

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

4 Zadanie nr 4 – wyszukiwanie z wykorzystaniem funkcji haszującej

Zaimplementuj kolejny algorytm wyszukiwania robota na liście.

1. Pobierz od użytkownika docelowy współczynnik wypełnienia tablicy α (przypomnienie: α to stosunek liczby wypełnionych elementów w tablicy, do długości tablicy). 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 2, 3, 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. W przypadku algorytmu z zadania nr 4, przedstaw wynik dla kilku wartości współczynnika α .

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

Wyjście: wykresy średniego czasu wyszukiwania robota.