Algorytmy i struktury danych - Problem Plecakowy

Aleksandra Ostrowska, nr 156121

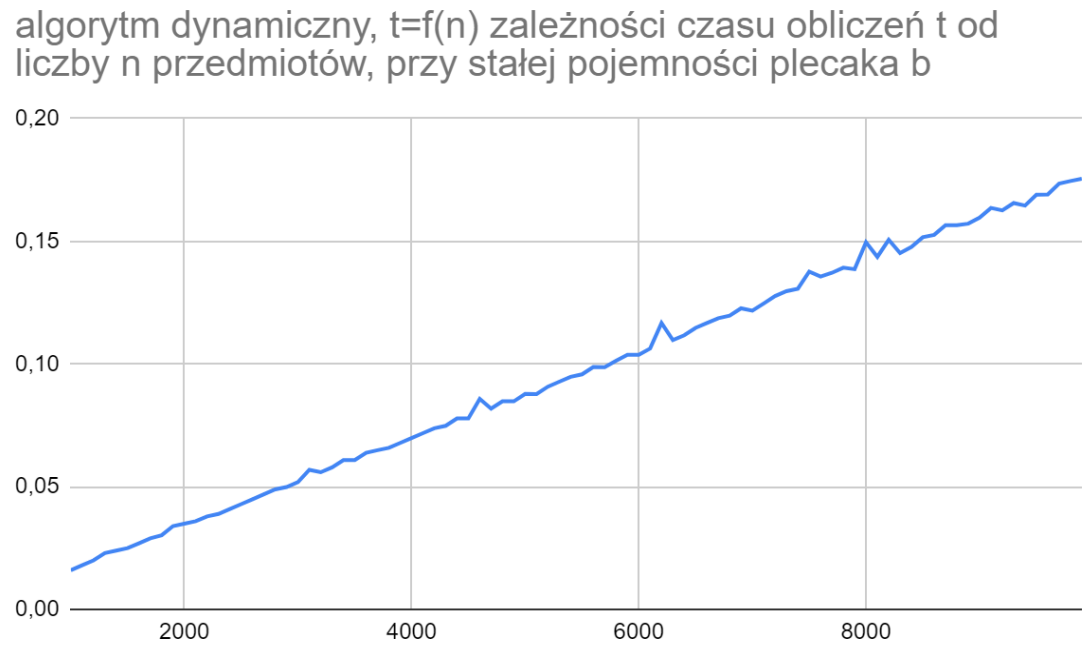
W tym sprawozdaniu będzie porównywana efektywność algorytmu zachłannego, przeszukiwania siłowego oraz algorytmu dynamicznego, w rozwiązywaniu problemu plecakowego.
Problem optymalizacyjny problemu plecakowego należy do klasy problemów NP-trudnych (w zwykłym sensie). Wersja decyzyjna zaś jest w klasie problemów NP-zupełnych.

Algorytm zachłanny działa poprzez wybieranie przedmiotów o największym stosunku wartości do wagi i dodawanie ich do plecaka, dopóki jest dostępne miejsce. Złożoność czasowa algorytmu zachłannego wynosi $O(n \log_2 n)$, gdzie n to liczba przedmiotów. Czas oczekiwania na rozwiązanie rośnie w sposób logarytmiczny wraz z liczbą przedmiotów.

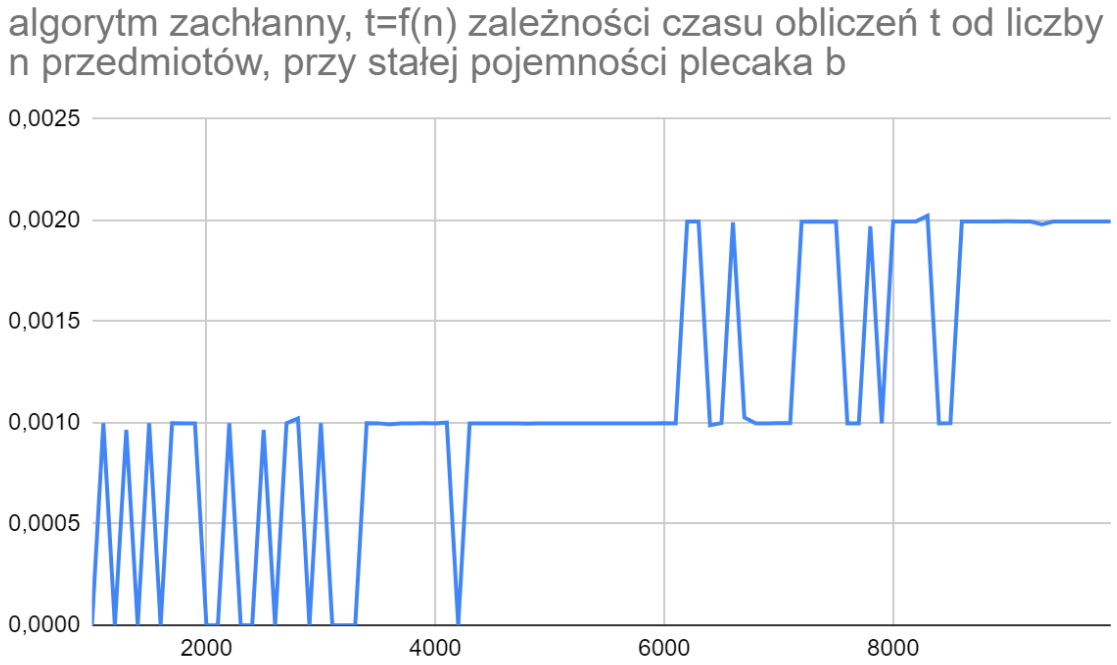
Dla problemu plecakowego, algorytm dynamiczny tworzy macierz o wymiarach (n+1)x(c+1), gdzie wartości w komórkach odpowiadają maksymalnej wartości plecaka dla danego przedmiotu i pojemności. Złożoność czasowa algorytmu programowania dynamicznego wynosi O(nc), gdzie n to liczba przedmiotów, a c to pojemność plecaka. Czas obliczeń rośnie liniowo wraz z liczbą przedmiotów i pojemnością plecaka. Kosztem większego czasu obliczeń algorytm programowania dynamicznego daje zawsze optymalne rozwiązanie w porównaniu do algorytmu zachłannego.

Dla algorytmu siłowego złożoność obliczeniowa wynosi O(2^n), gdzie n to liczba przedmiotów w plecaku. Wynika to z faktu, że algorytm generuje każde możliwe rozwiązanie, po czym wybiera optymalne. Znaczy to, że czas oczekiwania na rozwiązanie bardzo szybko rośnie w porównaniu do pozostałych algorytmów, jest to także widoczne na poniższych wykresach.

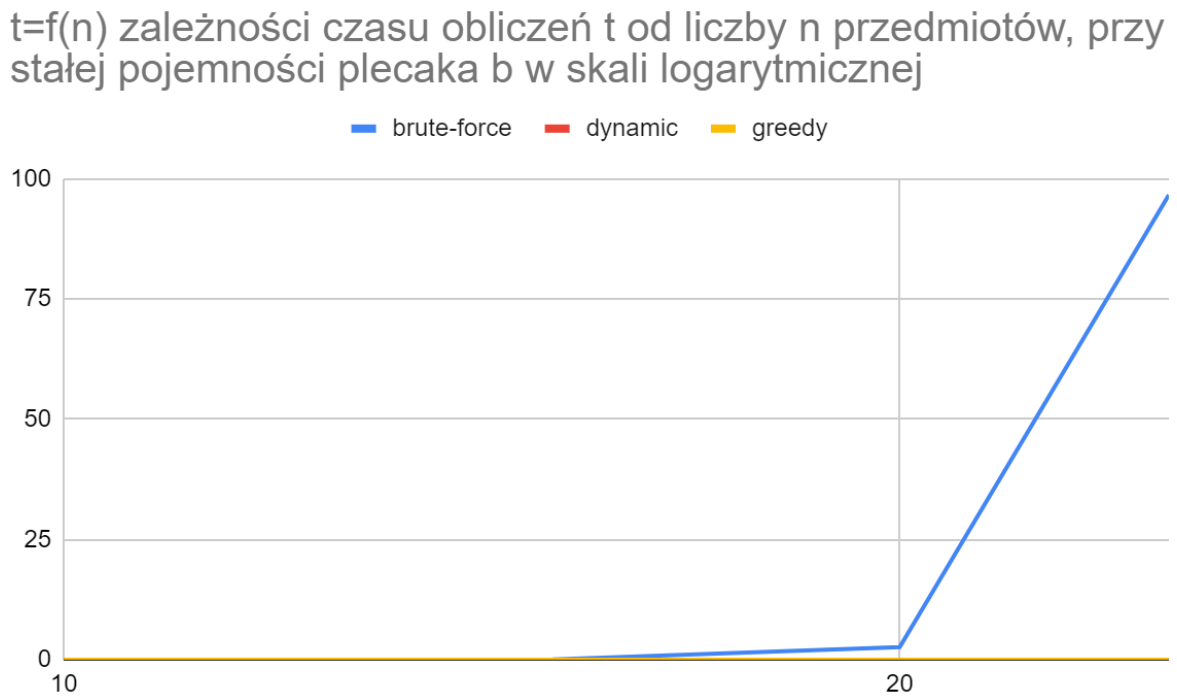
Rysunek 1: t=f(n) zależności czasu obliczeń t od liczby n przedmiotów, przy stałej pojemności plecaka b dla algorytmu siłowego (brute-force).



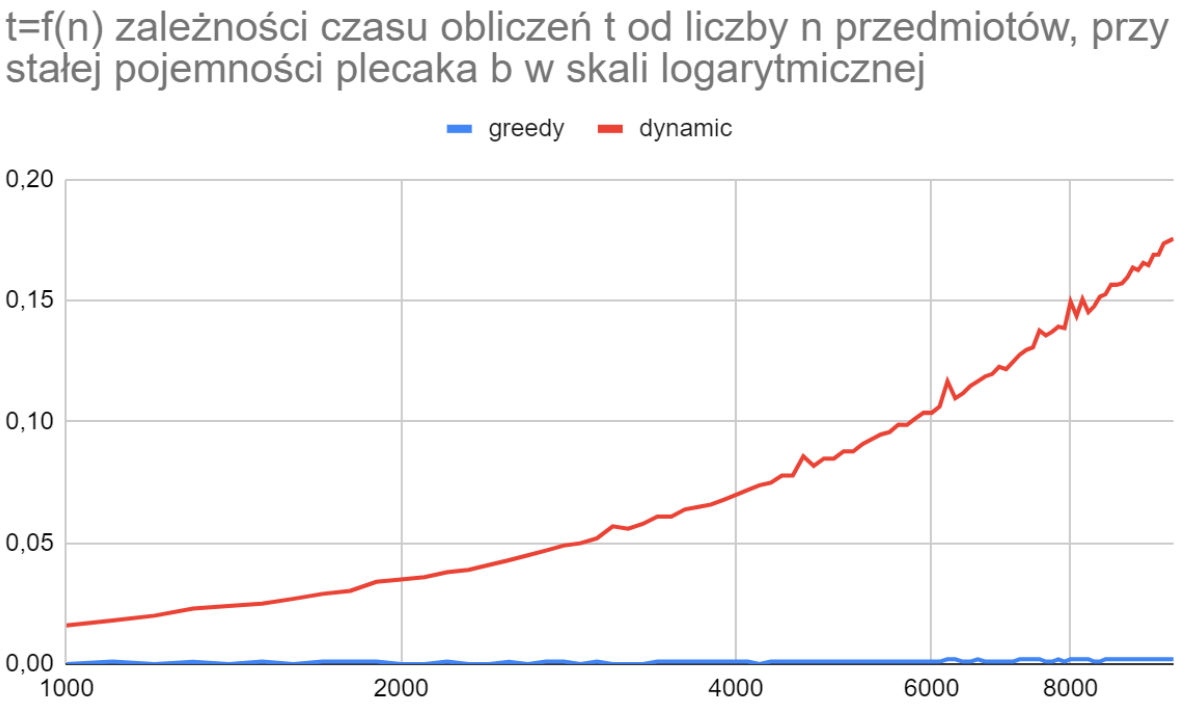
Rysunek 2: t=f(n) zależności czasu obliczeń t od liczby n przedmiotów, przy stałej pojemności plecaka b dla programowania dynamicznego.

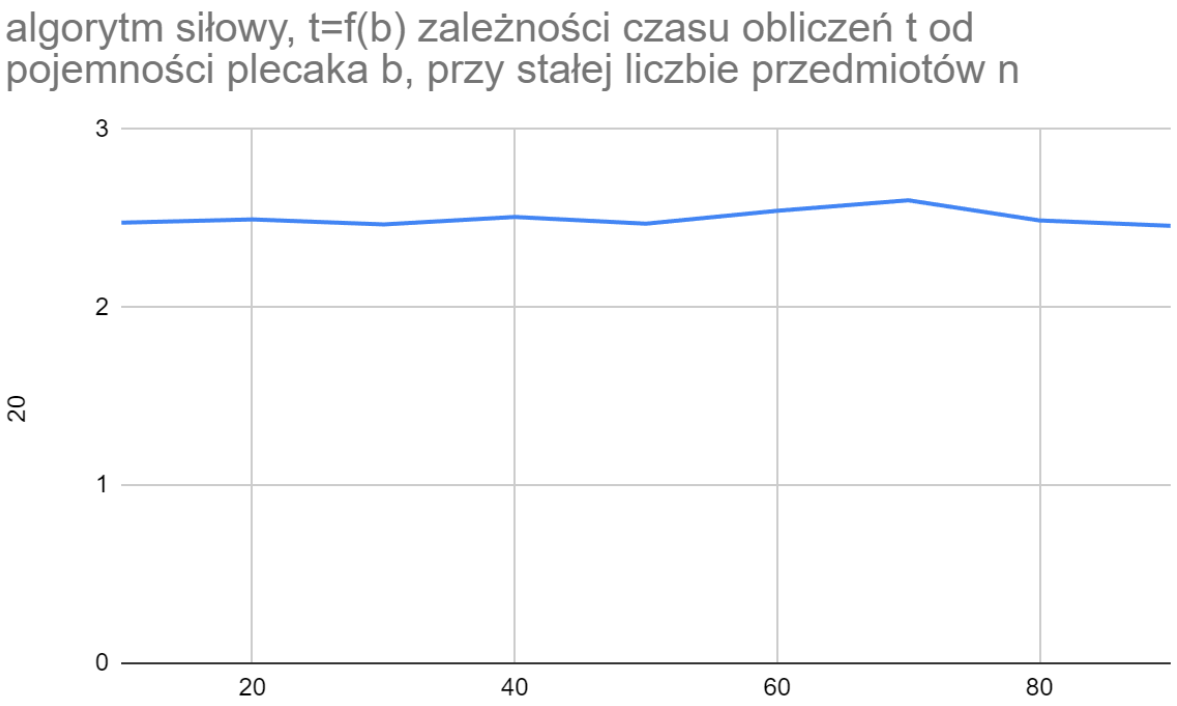


Rysunek 3: t=f(n) zależności czasu obliczeń t od liczby n przedmiotów, przy stałej pojemności plecaka b dla algorytmu zachłannego (greedy).

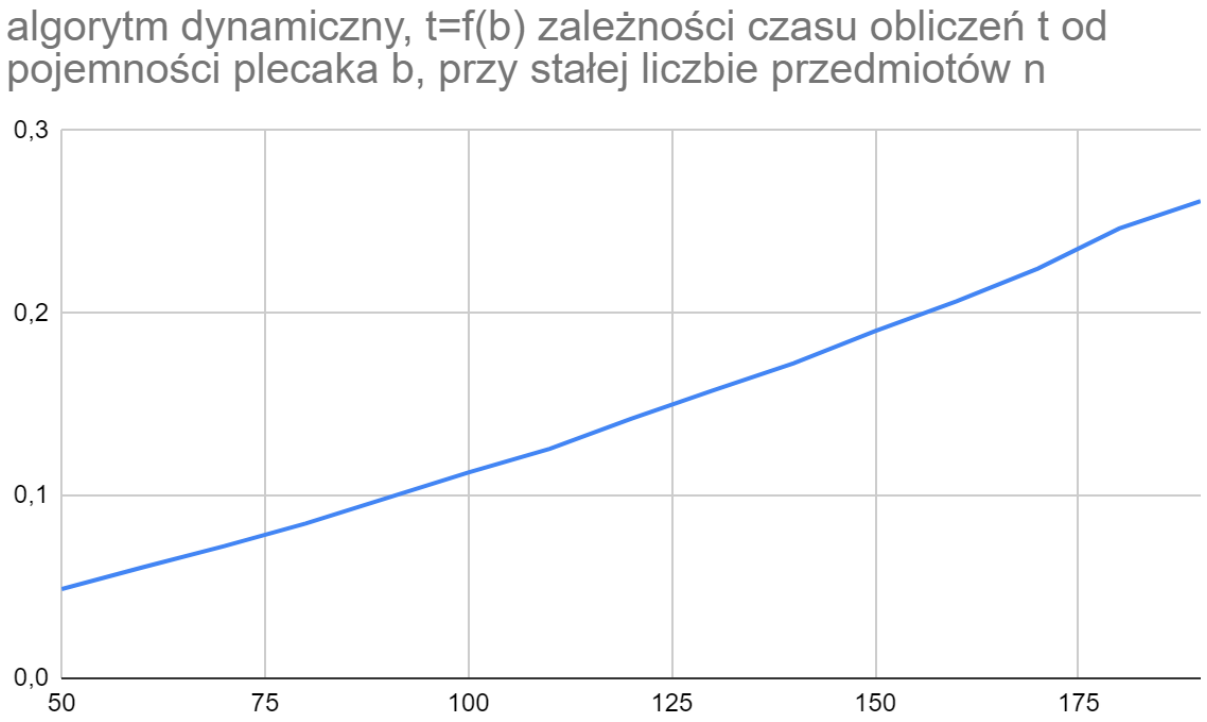


Rysunki 4 i 5: Zależność czasu obliczeń od pojemności plecaka (t=f(b)) przy stałej liczbie przedmiotów n. Zależność została przedstawiona na dwóch wykresach, aby ukazać różnice między algorytmem siłowym, a pozostałymi oraz efektywność programowania dynamicznego, a szybkością zachłannego.

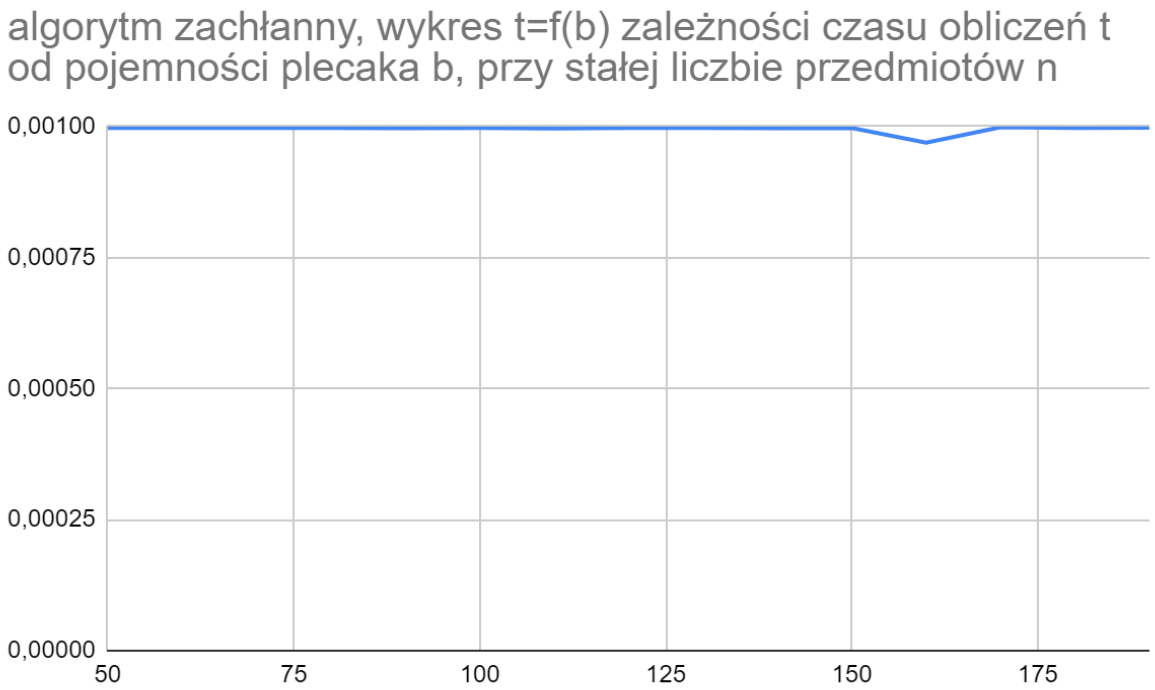




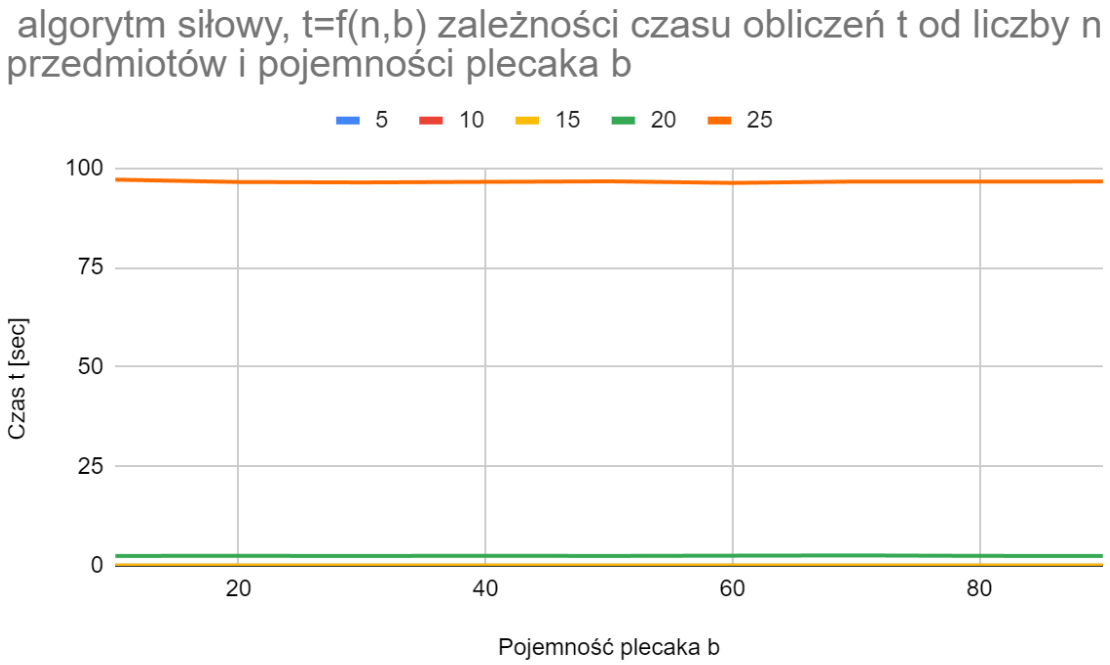
Rysunek 6: t=f(b) zależności czasu obliczeń t od pojemności plecaka b, przy stałej liczbie przedmiotów n dla algorytmu siłowego (brute-force).



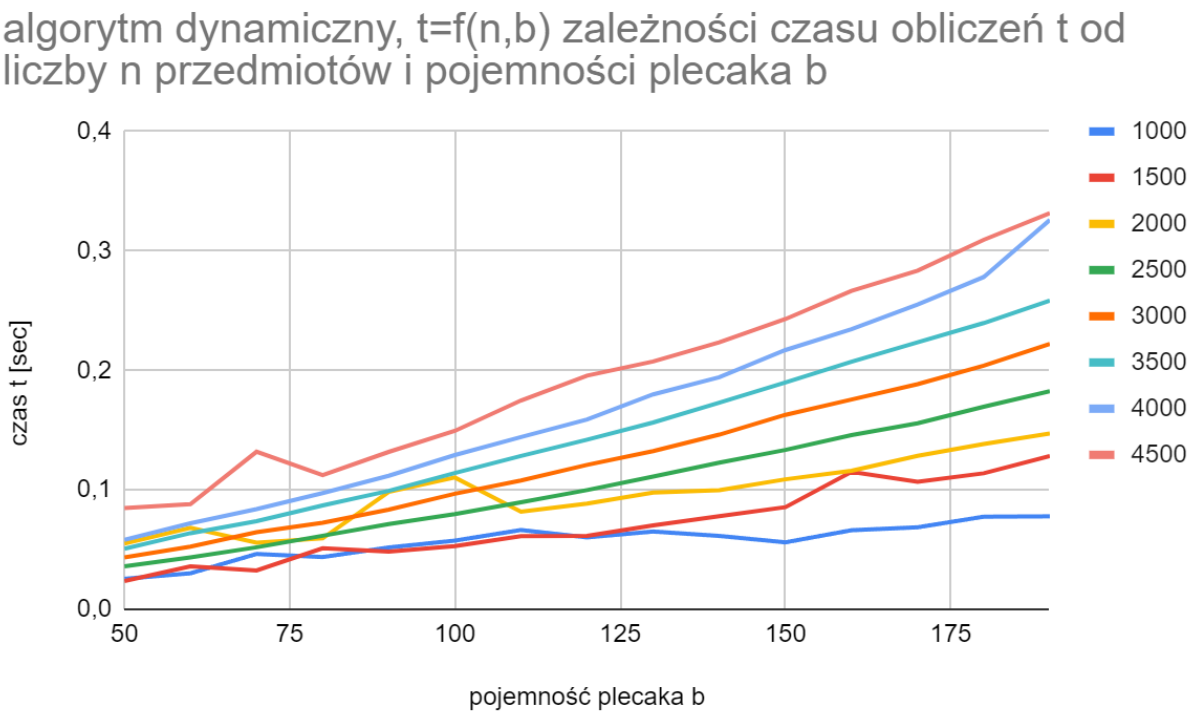
Rysunek 7: t=f(b) zależności czasu obliczeń t od pojemności plecaka b, przy stałej liczbie przedmiotów n dla programowania dynamicznego.



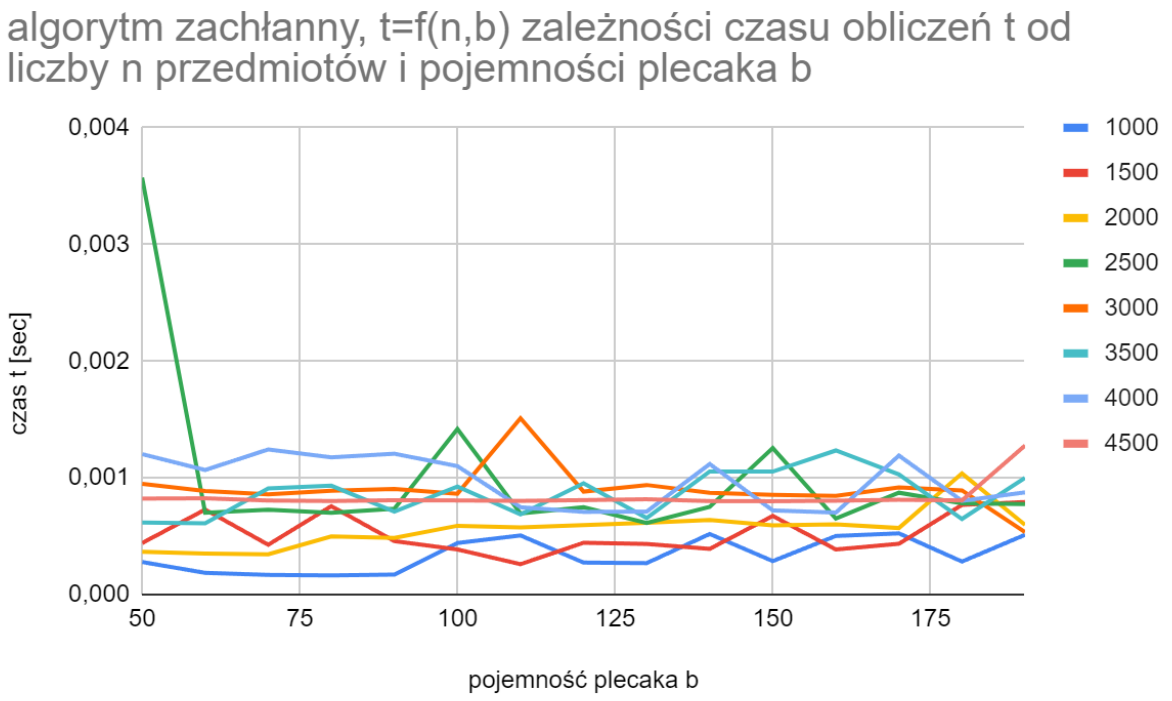
Rysunek 8: t=f(b) zależności czasu obliczeń t od pojemności plecaka b, przy stałej liczbie przedmiotów n dla algorytmu zachłannego (greedy).



Rysunek 9: t=f(n,b) zależności czasu obliczeń t od liczby n przedmiotów i pojemności plecaka b dla algorytmu siłowego (brute-force).



Rysunek 10: t=f(n,b) zależności czasu obliczeń t od liczby n przedmiotów i pojemności plecaka b dla programowania dynamicznego.



Rysunek 11: t=f(n,b) zależności czasu obliczeń t od liczby n przedmiotów i pojemności plecaka b dla algorytmu zachłannego (greedy).

Obserwacje na podstawie działania algorytmów:

Najmniej efektywny algorytm?

Algorytm siłowy szybko rośnie z każdym dodanym przedmiotem do plecaka, osiągając dla 25 elementów ponad 3000 sekund, gdzie dla pozostałych algorytmów taka wartość nie zajmuje nawet sekundy do spełnienia. Zmiana pojemności plecaka w tym algorytmie nie wpływa na długość jego wykonywania, co różni się od reszty algorytmów.

Stabilność programowania dynamicznego.

Algorytm programowania dynamicznego zgodnie z podaną w wstępie złożonością obliczeniową rośnie w sposób liniowy. Jedyną wadą algorytmu jest złożoność pamięciowa ze względu na macierz programowania dynamicznego, która przechowuje wyniki, z czego większość nie jest użyteczna. Mimo wszystko pod względem szybkość wykonywania i niezawodności algorytmu, który zawsze zwraca rozwiązanie optymalne, jest to najlepszy sposób na badanie problemu plecakowego.

Problem algorytmu zachłannego?

Ze względu na logarytmiczny wzrost czasu rozwiązania algorytmu zachłannego, na odpowiedź programu dla olbrzymich ilości przedmiotów do wyboru nie zabierze więcej niż 0,005 sekundy. Wynika to z prostoty rozwiązania owego problemu. Niestety, chociaż najbardziej efektywny czasowo, algorytm zachłanny podejmuje decyzje na podstawie lokalnie optymalnych wyborów w danym momencie, bez uwzględnienia długoterminowych konsekwencji. Jeśli istnieje zależność od kolejności, wtedy zachłanny algorytm może dać suboptymalne rozwiązanie. Przykładem jest problem komiwojażera, gdzie kolejność odwiedzania miast ma znaczenie, i algorytm zachłanny może prowadzić do znalezienia trasy, która jest krótsza od jednego punktu startowego, ale nie jest najkrótszą trasą ogółem.