

## Organizacja testów

---

Projektując układ testów warto zwrócić uwagę, aby testowanie było w miarę możliwości sprawne i wygodne. Proponowane jest takie podejście, aby po wyborze problemu (plecakowy lub komiwojażera) dla danych wczytanych z pliku wykonać od razu, po kolei, wszystkie zaimplementowane algorytmy z wypisaniem wyników na ekranie. Na przykład dla algorytmu plecakowego w wersji maksymalnej: przegląd zupełny, algorytm zachłanny i algorytm oparty na programowaniu dynamicznym. Umożliwi to łatwe przejrzanie i porównanie wyników uzyskanych dla tego samego problemu przy użyciu różnych algorytmów.

## Algorytm plecakowy

---

Wyniki powinny zawierać optymalny zestaw przedmiotów, z podaniem ich rozmiaru/wartości oraz podsumowanie dla wszystkich zapakowanych przedmiotów. Podsumowanie jest konieczne, aby możliwa była weryfikacja poprawności rozwiązania (sprawdzenie, czy suma rozmiarów nie przekracza pojemności plecaka i czy uzyskana suma wartości jest zgodna z oczekiwaną). Można również dodać indeks przedmiotu, który ułatwi znalezienie przedmiotu znajdującego się w rozwiązaniu na liście dostępnych przedmiotów.

### Problem

Capacity: 30

Items: 10

Item	Size	Value
0	9	116
1	13	103
8	9	43
9	6	115

### Solution

#### Full Search

Item	Size	Value
0	9	116
2	3	84
3	5	83
7	3	103
9	6	115
Total	26	501

Tutaj wyniki dla innych zaimplementowanych algorytmów, zależnie od wybranego zestawu.

## Algorytm komiwojażera

---

Wyniki powinny zawierać długość trasy (uwzględniającą również powrót do miasta początkowego) oraz sekwencję numerów kolejno odwiedzanych miast.

### Problem

Number of cities: 6

	0	1	2	3	4	5
0	0	20	30	31	28	40
1	30	0	10	14	20	44
5	50	54	60	52	38	0

### Solution

#### Full Search

Length = 132

Path = 0 - 1 - 2 - 3 - 4 - 5 - 0

Tutaj wyniki dla innych zaimplementowanych algorytmów, zależnie od wybranego zestawu.