

Algorytmy i struktury danych, Teleinformatyka, I rok

Raport z laboratorium nr: 9

Imię i nazwisko studenta: Mateusz Warzecha

nr indeksu: 410846

1. W pole poniżej wklej najważniejszy (według Ciebie) fragment kodu źródłowego z zajęć (maksymalnie 15 linii).

```
for item in items:
    fit = check_fit(item.get_width(), item.get_height(), backapck)
    if fit:
        if (fit[2]):
            for x in range(fit[0], item.get_width() + fit[0]):
                for y in range(fit[1], item.get_height() + fit[1]):
                    backapck[y][x] = item.get_id()
        if (fit[2]):
            for x in range(fit[0], item.get_width() + fit[0]):
                for y in range(fit[1], item.get_height() + fit[1]):
                    backapck[y][x] = item.get_id()
        else:
            for x in range(fit[0], item.get_width() + fit[0]):
                for y in range(fit[1], item.get_height() + fit[1]):
                    backapck[x][y] = item.get_id()
    value += item.get_value()
```

Wybrałem ten fragment kodu ,ponieważ zawiera się w nim kluczowa do działania algorytmu funkcja wkładająca poszczególne elementy do plecaka oraz wywołanie również istotnej funkcji check_fit która sprawdza czy dany element się zmieści w plecaku.

```
size = 20
time = 0.0029761791229248047
reached value = 131
max possible value = 161
```

```
size = 100
time = 1.6055245399475098
reached value = 2388
max possible value = 2653
```

```
size = 500
time = 1081.4609174728394
reached value = 62240
max possible value = 68634
```

Jak widać na załączonych zdjęciach algorytm naiwny już dla plecaka o wymiarach 500x500 miał ogromne problemy z wydajnością. Ponadto nie poradził on sobie tak źle, ponieważ sortując elementy po ich stosunku wartości do powierzchni zazwyczaj jest to od 80%-90% całkowitej wartości przedmiotów.