

# Rucksack Problem Algorithmus

Rucksack Problem oder auch "Knapsack Problem".  
Am Rucksack Problem kann man die Vorteile des Backtrackings aufzeigen, da hier das globale Optimum gesucht wird.

## Beispiel Backtracking

### Quellcode

```
1. public static void main(String[] args) {
2. int[] werte = new int{5,4,3,2,1};
3. int[] gewichte = new int{10,5,20,30,15};
4. }
5.
6. public static int rucksack(int i, int kap, int wert) {
7. if ((i >= n) || (kap <= 0))
8. return wert;
9. if (gewichte[i] > kap)
10. return rucksack(i+1,kap,wert);
11. int mit=rucksack(i+1, kap-gewichte[i], wert+werte[i]);
12. int ohne=rucksack(i+1, kap, wert);
13. return max(mit, ohne);
14. }
```

Alles anzeigen

## Weblinks

- [www-i1.informatik.rwth-aachen.de/~algorithmus/algo15.php](http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo15.php)

## Inhaltsverzeichnis

- [1 Beispiel Backtracking](#)
- [2 Weblinks](#)