

Dynamic Programming

Table of Contents

Knapsack: bottom up

```
1 function KnapSack(v[1..n], w[1..n], W):
2   # initialise first column with 0
3   for i from 0 to n:
4     K[i, 0] = 0
5
6   # initialise first row with 0
7   for j from 1 to W:
8     K[0, j] = 0;
9
10  # iterate over table
11  for i from 1 to n:
12    for j from 1 to W:
13      if j < w[i]:
14        K[i, j] = K[i-1, j]
15      else:
16        K[i, j] = max(K[i-1, j], K[i-1, j-w[i]] + v[i])
17
18  return K[n,W]
```

Breadth first search