# KnapSack.java

```
   package com.example.dynamicProgramming;

   public class KnapSack {
       public int max(int a, int b) {
3          return (a > b) ? a : b;
       }

       public int knapSack1(int W, int wt[], int val[], int n) {
           // Base Case
2          if (n == 0 || W == 0)
               return 0;

3          if (wt[n - 1] > W)
2              return knapSack1(W, wt, val, n - 1);

           else
5              return max(val[n - 1]
2                      + knapSack1(W - wt[n - 1], wt,
                               val, n - 1),
                       knapSack1(W, wt, val, n - 1));
       }

       // Returns the value of maximum profit
       public int knapSackRec(int W, int wt[], int val[],
               int n, int[][] dp) {

           // Base condition
2          if (n == 0 || W == 0)
               return 0;

1          if (dp[n][W] != -1)
1              return dp[n][W];

3          if (wt[n - 1] > W)

               // Store the value of function call
               // stack in table before return
2              return dp[n][W] = knapSackRec(W, wt, val, n - 1, dp);

           else

               // Return value of table after storing
5              return dp[n][W] = max((val[n - 1]
2                      + knapSackRec(W - wt[n - 1], wt, val,
                               n - 1, dp)),
                       knapSackRec(W, wt, val, n - 1, dp));
       }

       public int knapSack2(int W, int wt[], int val[], int N) {

           // Declare the table dynamically
52  2          int dp[][] = new int[N + 1][W + 1];
53
54             // Loop to initially filled the
55             // table with -1
56  3          for (int i = 0; i < N + 1; i++)
57  3              for (int j = 0; j < W + 1; j++)
58                     dp[i][j] = -1;
59
60  1          return knapSackRec(W, wt, val, N, dp);
61         }
62
63         public int knapSack3(int W, int wt[], int val[], int n) {
64             int i, w;
65  2          int K[][] = new int[n + 1][W + 1];
66
67             // Build table K[][] in bottom up manner
68  2          for (i = 0; i <= n; i++) {
69  2              for (w = 0; w <= W; w++) {
70  2                  if (i == 0 || w == 0)
71                         K[i][w] = 0;
72  3                  else if (wt[i - 1] <= w)
73  6                      K[i][w] = max(val[i - 1]
74                                 + K[i - 1][w - wt[i - 1]],
75                                 K[i - 1][w]);
76                     else
77  1                      K[i][w] = K[i - 1][w];
78                 }
79             }
80
81  1          return K[n][W];
82         }
83
84         public int knapSack4(int W, int wt[], int val[], int n) {
85             // Making and initializing dp array
86  1          int[] dp = new int[W + 1];
87
88  3          for (int i = 1; i < n + 1; i++) {
89  2              for (int w = W; w >= 0; w--) {
90
91  3                  if (wt[i - 1] <= w)
92
93                         // Finding the maximum value
94  4                      dp[w] = Math.max(dp[w], dp[w - wt[i - 1]]
95                                 + val[i - 1]);
96                 }
97             }
98             // Returning the maximum value of knapsack
99  1          return dp[W];
100        }
101
102    }
```

## Mutations

```
5      1. negated conditional → KILLED
```

| | |
|---|---|
| | 2. changed conditional boundary → SURVIVED<br>3. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::max → KILLED |
| 10 | 1. negated conditional → KILLED<br>2. negated conditional → KILLED |
| 13 | 1. Replaced integer subtraction with addition → KILLED<br>2. negated conditional → KILLED<br>3. changed conditional boundary → KILLED |
| 14 | 1. Replaced integer subtraction with addition → KILLED<br>2. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSack1 → SURVIVED |
| 17 | 1. Replaced integer subtraction with addition → KILLED<br>2. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSack1 → KILLED<br>3. Replaced integer subtraction with addition → KILLED<br>4. Replaced integer subtraction with addition → KILLED<br>5. Replaced integer subtraction with addition → KILLED |
| 18 | 1. Replaced integer addition with subtraction → KILLED<br>2. Replaced integer subtraction with addition → KILLED |
| 28 | 1. negated conditional → KILLED<br>2. negated conditional → KILLED |
| 31 | 1. negated conditional → KILLED |
| 32 | 1. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSackRec → NO_COVERAGE |
| 34 | 1. Replaced integer subtraction with addition → KILLED<br>2. changed conditional boundary → KILLED<br>3. negated conditional → KILLED |
| 38 | 1. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSackRec → SURVIVED<br>2. Replaced integer subtraction with addition → KILLED |
| 43 | 1. Replaced integer subtraction with addition → KILLED<br>2. Replaced integer subtraction with addition → KILLED<br>3. Replaced integer subtraction with addition → KILLED<br>4. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSackRec → KILLED<br>5. Replaced integer subtraction with addition → KILLED |
| 44 | 1. Replaced integer addition with subtraction → KILLED<br>2. Replaced integer subtraction with addition → KILLED |
| 52 | 1. Replaced integer addition with subtraction → KILLED<br>2. Replaced integer addition with subtraction → KILLED |
| 56 | 1. negated conditional → KILLED<br>2. changed conditional boundary → KILLED<br>3. Replaced integer addition with subtraction → KILLED |
| 57 | 1. Replaced integer addition with subtraction → KILLED<br>2. negated conditional → KILLED<br>3. changed conditional boundary → KILLED |
| 60 | 1. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSack2 → KILLED |
| 65 | 1. Replaced integer addition with subtraction → KILLED<br>2. Replaced integer addition with subtraction → KILLED |
| 68 | 1. negated conditional → KILLED<br>2. changed conditional boundary → KILLED |
| 69 | 1. changed conditional boundary → KILLED<br>2. negated conditional → KILLED |
| 70 | 1. negated conditional → KILLED<br>2. negated conditional → KILLED |
| 72 | 1. negated conditional → KILLED<br>2. changed conditional boundary → KILLED<br>3. Replaced integer subtraction with addition → KILLED |
| 73 | 1. Replaced integer subtraction with addition → KILLED<br>2. Replaced integer subtraction with addition → KILLED<br>3. Replaced integer subtraction with addition → KILLED<br>4. Replaced integer addition with subtraction → KILLED<br>5. Replaced integer subtraction with addition → KILLED |

| | |
|---|---|
| | 6. Replaced integer subtraction with addition → KILLED |
| 77 | 1. Replaced integer subtraction with addition → KILLED |
| 81 | 1. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSack3 → KILLED |
| 86 | 1. Replaced integer addition with subtraction → KILLED |
| 88 | 1. Replaced integer addition with subtraction → KILLED<br>2. negated conditional → KILLED<br>3. changed conditional boundary → KILLED |
| 89 | 1. negated conditional → KILLED<br>2. changed conditional boundary → SURVIVED |
| 91 | 1. negated conditional → KILLED<br>2. Replaced integer subtraction with addition → KILLED<br>3. changed conditional boundary → KILLED |
| 94 | 1. Replaced integer subtraction with addition → KILLED<br>2. Replaced integer subtraction with addition → KILLED<br>3. Replaced integer addition with subtraction → KILLED<br>4. Replaced integer subtraction with addition → KILLED |
| 99 | 1. replaced int return with 0 for com/example/dynamicProgramming/KnapSack::knapSack4 → KILLED |

## Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

## Tests examined

- com.example.dynamicProgramming.KnapSackTest.test4(com.example.dynamicProgramming.KnapSackTest) (0 ms)
- com.example.dynamicProgramming.KnapSackTest.test2(com.example.dynamicProgramming.KnapSackTest) (0 ms)
- com.example.dynamicProgramming.KnapSackTest.test3(com.example.dynamicProgramming.KnapSackTest) (0 ms)
- com.example.dynamicProgramming.KnapSackTest.test(com.example.dynamicProgramming.KnapSackTest) (0 ms)

Report generated by PIT 1.15.0