

Implement 0/1 Knapsack problem
using dynamic programming

413

include <stdio.h>

int max (int a, int b) {return (a > b) ? a : b;}

int Knapsack (int w, int wt[], int val[],
int n)

{

int i, w;

int k[n+1][w+1];

for (i = 0; i <= n; i++)

{
for (w = 0; w <= w; w++){
for (w = 0; w <= w; w++){
if (i == 0 || w == 0)

k[i][w] = 0;

else if (wt[i-1] <= w)

(1)

$$K[i][w] = \max(\text{val}[i-1] + K[i-1][w-w[i]], K[i-1][w]);$$

else
 $K[i][w] = K[i-1][w];$

$\}$
 $\}$
 return $K[n][w];$

$\}$
 int main ()

$\{$
 int i, n, val[20], wt[20], w;

printf ("Enter no. of items:");

scanf ("%d", &n);

printf ("Enter value and weight of items: \n");

for (i = 0; i < n; ++i) {

scanf ("%d %d", &val[i], &wt[i]);

}

printf ("Enter size of knapsack:");

scanf ("%d", &w);

printf ("%d", knapsack(w, wt, val, n));

return 0; }

②