

Modification of Knapsack

Md. Arman

Jaiye

: Give the count of items selected

IBM19CS089

4B

Code: #include <stdio.h>

int max (int, int);

int m, i, j, n, P[10], W[10], V[10][10], X[10], sz

soln - flag = 0;

int knapsack ();

void objed_selected ();

void main () {

printf ("Enter no. of objects \n");

scanf ("%d", &n);

printf ("Enter weights of N objects \n");

for (i = 1; i <= n; i++)

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

printf ("Enter profits of N objects \n");

for (i = 1; i <= n; i++)

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

printf ("Enter the capacity of knapsack \n");

scanf ("%d", &m);

(1)

```
dp_soln = knapsack (n, w, m, v, p);
```

```
printf ("Output is \n");
```

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

```
for (j=0; j<=m; j++) {
```

```
printf ("val\t", v[i][j]);
```

```
}
```

```
printf ("\n");
```

```
} printf ("Optimal solution = val\t", dp_soln);
```

```
obj_selected ();
```

```
}
```

```
int max (int a, int b) {
```

```
return (a > b ? a : b);
```

```
}
```

```
int knapsack () {
```

```
int i, j;
```

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

```
for (j=0; j<=m; j++) {
```

```
if (i==0 || j==0)
```

```
v[i][j] = 0;
```

2

else {

if ($w[i] > j$)

$v[i][j] = v[i-1][j];$

else

$v[i][j] = \max(v[i-1][j], v[i-1]$

$[j - w[i]] + p[i]);$

}

return $v[n][m];$

} void object_selected () {

$i = n;$

$j = m;$

while ($i \neq 0 \ \& \ j \neq 0$) {

if ($v[i][j] \neq v[i-1][j]$) {

$x[i] = 1;$

$j = j - w[i];$

}

$i--;$

} print ("objects selected \n");

```
for (i = 1; i <= n; i++) {
```

```
    if (x[i] == 1) {
```

```
        flag++;
```

```
        printf ("%d \t", i);
```

```
    }
```

```
}
```

```
printf ("Count of items selected : %d",  
        flag);
```

```
{
```