

# 0929\_ASStudy4주차\_오윤기

3번 12865 평범한배낭

```
public class Main_백준12865_평범한배낭 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int W = sc.nextInt();
        int[] weights = new int[N + 1];
        int[] values = new int[N + 1];
        for (int i = 1; i <= N; i++) {
            weights[i] = sc.nextInt();
            values[i] = sc.nextInt();
        }

        int[][] DP = new int[N + 1][W + 1];
        for (int i = 1; i <= N; i++) {
            for (int w = 1; w <= W; w++) {
                if (weights[i] <= w) { //가방에 넣을 수 있을 때
                    DP[i][w] = Math.max(DP[i - 1][w], values[i] + DP[i - 1][w - weights[i]]);
                    //해당 물건을 넣고 이전상태에서 남은무게의 가치의 합과 이전상황 무게에서의 가치 합 중 큰 것을 선택한다.
                } else { // 가방에 넣을 수 없을 때
                    DP[i][w] = DP[i - 1][w];
                }
            }
        }
        for(int i = 1; i<=N; i++) {
            System.out.println(i + "까지 넣었을 때의 상황: " +Arrays.toString(DP[i]));
        }
        System.out.println(DP[N][W]);
        sc.close();
    }
}
```

2차원 배열 사용한 버전

```
public class Main_백준12865_평범한배낭2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int W = sc.nextInt();

        int[] weights = new int[N + 1];
        int[] profits = new int[N + 1];
```

```

for (int i = 1; i <= N; i++) {
    weights[i] = sc.nextInt();
    profits[i] = sc.nextInt();
}

int[] D = new int[W + 1];
// 가치판단할 때만 유리 , 어떤물건 있는지는 확인 함
for (int i = 1; i <= N; i++) {
    for (int w = W; w >= weights[i]; w--) {
        D[w] = Math.max(D[w], profits[i] + D[w - weights[i]]);
    }
    System.out.println(i + "까지 넣었을 때의 상황: " + Arrays.toString(D));
}
System.out.println(D[W]);
sc.close();
}
}

```

1차원 배열을 사용한 버전