

1. unbounded knapsack problem

```
public class UnboundedKnapsackProblem {  
  
    private static int unboundedKnapsack(int W, int n, int[] val,  
int[] wt) {  
  
        int dp[] = new int[W + 1];  
  
        for (int i = 0; i <= W; i++) {  
            for (int j = 0; j < n; j++) {  
                if (wt[j] <= i) { //물건의 무게가 배낭무게를  
//넘을수 없게 설정  
  
                    dp[i] = Math.max(dp[i], dp[i - wt[j]]  
+ val[j]); // 점화식  
                    System.out.print(dp[i] + " ");  
                }  
            }  
            System.out.println();  
        }  
        return dp[W]; //결과를 출력해줌  
    }  
  
    public static void main(String[] args) {  
        int W = 10; //배낭의크기 초기화  
        int value[] = { 100000, 400000, 300000, 500000 };  
        //각각의 가치를 초기화  
        int w[] = { 5, 4, 6, 3 }; //각각의 무게를 초기화  
        int n = value.length; // 물건의 갯수 초기화  
        System.out.println(unboundedKnapsack(W, n, value, w));  
    } // 각각의 값을 초기화  
}
```

```
500000  
400000 500000  
100000 400000 500000  
100000 400000 400000 1000000  
100000 900000 900000 1000000  
600000 900000 900000 1000000  
600000 900000 900000 1500000  
600000 1400000 1400000 1500000  
1500000
```

DP표 출력이 좀 이상합니다 ㅠ

배낭 무게 10에 관한 unbounded knapsack problem은 무게가 3, 가치가 500000인 물건이 3개가 들어가면 되어 1500000 이라는 값이 출력됩니다.

## 2. 0/1 knapsack problem

```
class dp {
    int weight;
    int value;

    dp(int weight, int value) {
        this.weight = weight;
        this.value = value;
    }
} // 값을 초기화해줌

public class KnapsackProblem01 {
    public KnapsackProblem01() {
        System.out.println("물건의 갯수를 입력해주세요 : ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(); // 물건의 갯수를 입력 받음
        System.out.println("가방의 용량을 입력해주세요 : ");
        int w = sc.nextInt(); // 배낭의 최대 무게를 입력 받음
        dp DP[] = new dp[n + 1]; // 1부터 시작하기 위함
        System.out.println("물건의 무게와 가치를 입력해주세요 : ");
        for (int i = 1; i <= n; i++)
            DP[i] = new dp(sc.nextInt(), sc.nextInt()); //물건
        // 각각의 무게와 가치를 입력 받음
        int temp[][] = new int[n + 1][w + 1]; // 1부터 시작하기
        // 위함

        for (int i = 1; i < temp.length; i++) {
            for (int j = 1; j < temp[i].length; j++) {
                if (DP[i].weight > j)
                    System.out.print((temp[i][j] = temp[i - 1][j]) + " "); //값이 없으면 그대로 0으로 출력
                else {
                    System.out.print((
                        temp[i][j] =
                        Math.max(DP[i].value + temp[i - 1][j - DP[i].weight], temp[i - 1][j])) + " ");
                    // 값이 있으면 그 값을 출력
                }
            }
            System.out.println();
        }
        System.out.println(temp[n][w]); //최종 답안 출력
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        new KnapsackProblem01(); //위에 코드를 돌리기
    }
}
```

}

}

```
물건의 갯수를 입력해주세요 :
4
가방의 용량을 입력해주세요 :
10
물건의 무게와 가치를 입력해주세요 :
5 100000
4 400000
6 300000
3 500000
0 0 0 0 100000 100000 100000 100000 100000 100000
0 0 0 400000 400000 400000 400000 400000 500000 500000
0 0 0 400000 400000 400000 400000 400000 500000 700000
0 0 500000 500000 500000 500000 900000 900000 900000 900000
900000
```

물건의 갯수를 입력하고 가방의용량을 입력합니다.

그 다음 그 물건의 각각의 무게와 가치를 입력합니다. 0/1배낭문제는 한 개의 물건의 2개 들어갈 수 없기 때문에 무게 3과 4가 들어간 무게7에 가치 900000이 최대가치가 됩니다.