

Laboratory work # 1

Student: HU Riqian

Student ID: 20321114

Timus Name: hduads2022_20321114

Mail: jhlxhrq@163.com

Problem # 1005. Stone Pile

Screenshot from Timus:

9796482	11:49:46 29 Mar 2022	hduads2022_20321114	1005. Stone Pile	Java 1.8	Accepted		0.203	5 068 KB
---------	-------------------------	-------------------------------------	----------------------------------	----------	----------	--	-------	----------

Explanation of algorithm:

1. We can regard the question as 0-1 Knapsack.
2. Consider using dynamic programming to solve the problem.

Computational complexity of algorithm:

$$O(Vn)$$

Source code:

```
import java.util.Scanner;
import java.lang.Math;

public class StonePile {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        /**
         * n : Number of the stones.
         * w : Array of the stones. n = w.length;
         * s : Total weight of the stones.
         * m : Half of the total weight, which is the target weight of the
         classification of the these stones.
         */
        int n = in.nextInt();
        int[] w = new int[n];
        int s = 0;

        for (int i = 0; i < n; i++) {
            w[i] = in.nextInt();
            s += w[i];
        }

        // TO DO: Classify the stones into 2 groups, making one part be close
        to half of the total
        // weight, then make the difference of the 2 groups.
    }
}
```

```
int m = s - s / 2;
int[] f = new int[m + 1];

// We use the method of 0-1 Knapsack.
for (int i = 0; i < n; i++) {
    for (int j = m; j > w[i] - 1; j += -1) {
        f[j] = Math.max(f[j], f[j - w[i]] + w[i]);
    }
}

System.out.println(Math.abs(s - f[m] * 2));
}
```