

# Assignment 1.2: BFS and DFS

Aleksandra Ristic  
arc19002@student.mdh.se

## 1 Explanation of the problem

### a. Representation

Set of variables:

$X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$  -  $n$  is number of items

Domain of each variable:

$x_i = \{0, 1\}$  - if item  $i$  is chosen to be in bag(solution) or not

### b. Objective function

Goal is to select the items in the way to maximize profit with respect to the capacity.

$$\sum_{i=1}^n p_i x_i$$

$p_i$  represents profit of item  $i$

$n$  is number of item in knapsack and value of  $x_i$  depends of information does that item is selected or not.

### c. Restrictions

$$\sum_{i=1}^n w_i x_i \leq c$$

$w_i$  represents weight of item  $i$

$c$  is capacity of knapsack

$n$  is number of item in knapsack

## 2 Comparison of algorithms

### a. Time

On the figure 1. you can see difference between time needed for executing BFS and DFS. We can conclude that DFS is working faster than BFS, reason for that is that its need to pass throw all possible solutions in order to get the best combination.

```
0 BFS: 0.0058195 seconds
1
2
3 DFS: 0.001814 seconds
4
5 D:\D-participia\SANJA\VTIL-groden\AT\1
```

Figure 1: Comparison of the time

## b. Memory

On the figure 2. is shown memory usage by both algorithm. We can see that BFS need more memory (+0.98KB) then DFS (+0.48KB). DFS is using stack and its size depends of depth while the used memory of BFS's queue depends of width of graph.

Summary Events Memory Usage CPU Usage					
📷 Take Snapshot 🔍 View Heap ✖ Delete					
ID	Time	Objects (Diff)	Heap Size (Diff)		
1	0.16s	850 ( n/a )	103.17 KB ( n/a )		
2	0.17s	876 (+26 ↑)	104.15 KB (+0.98 KB ↑)	←	BFS
3	0.17s	881 (+5 ↑)	105.41 KB (+1.26 KB ↑)		
➤ 5	0.17s	892 (+11 ↑)	105.89 KB (+0.48 KB ↑)	←	DFS

Figure 2: Comparison of the memory.