Contents

[Introduction 2](#_Toc530924562)

[Algorithm 1 – Brute force 2](#_Toc530924563)

[Brief explanation 2](#_Toc530924564)

[Algorithm in psudo code 2](#_Toc530924565)

[Algorithm 2 – Dynamic Button up 2](#_Toc530924566)

[Brief explanation 2](#_Toc530924567)

[Algorithm in pseudo code 2](#_Toc530924568)

[Algorithm 3 – Top down 2](#_Toc530924569)

[Brief explanation 2](#_Toc530924570)

[Algorithm in psudo code 2](#_Toc530924571)

[Test case explanation 2](#_Toc530924572)

[How generate the sample test cases 2](#_Toc530924573)

[Testing environment 2](#_Toc530924574)

[How isolated we ran the test cases in 2](#_Toc530924575)

[Result of execution 2](#_Toc530924576)

[Comparing the result in graph possibly by time (running time comparison) 2](#_Toc530924577)

[References ( if we have any) 2](#_Toc530924578)

# Introduction

# Algorithm 1 – Brute force

## Brief explanation

## Algorithm in psudo code

# Algorithm 2 – Dynamic Button up

## Brief explanation

## Algorithm in pseudo code

# Algorithm 3 – Top down

## Brief explanation

## Algorithm in psudo code

# Test case explanation

## How generate the sample test cases

## Testing environment

## How isolated we ran the test cases in

# Result of execution

## Comparing the result in graph possibly by time (running time comparison)

# References ( if we have any)

Details dynamic programming lectures

<http://www.es.ele.tue.nl/education/5MC10/Solutions/knapsack.pdf>

<http://cse.unl.edu/~goddard/Courses/CSCE310J/Lectures/Lecture8-DynamicProgramming.pdf>