**Counting and Optimisation**

**Document Id 12**

Top Level Learning Goals

1 Understand Counting as it applies to Algorithms

2 Understand Optimisation

3 Understand associated concepts related to Counting and Optimisation such as

dynamic programming

**Section 1**

**Counting Concepts**

Learning Objectives

* Dynamic programming
* Sliding window

**Concepts**

* Counting for algorithms

https://en.wikipedia.org/wiki/Combinatorics

* Optimisation

https://en.wikipedia.org/wiki/Mathematical\_optimization

* Dynamic programming

https://en.wikipedia.org/wiki/Dynamic\_programming

**Guided Learning Task**

**Sliding window maxima and minima**

Generally speaking a sliding window is a sub-list that runs over an

underlying collection

**Sequence -> [a b c d e f g h] and w = 3**

**[a b c]**

**[b c d]**

**[c d e]**

**[d e f]**

**[e f g]**

**[f g h]**

Lab 1 Practical Application of a Sliding window

Lab 1 Solution

**Section 2**

**Counting patterns**

Learning Objectives

* Counting as it applies to Algorithms
* Combinations and Permutations

<https://en.wikipedia.org/wiki/Combination>

https://en.wikipedia.org/wiki/Permutation

Issues with large numbers and arithmetic opperations

Guided Learning Task

* Count Combinations
* Count Permutations

Lab 2 Practical Application of Counting to Permutations

Lab 2 Solution

**Section 3**

**Counting Theorem for Algorithms**

Learning Objectives

* Efficient Code

the fastest, most memory efficient code

as a rule the fewer instructions a program executes the faster it

runs

* Counting Theorem

https://en.wikipedia.org/wiki/Pigeonhole\_principle

Guided Learning Tasks

* Counting Sort
* Pigeon-Hole Sort

Lab 3 Practical Application of the Counting Theorem

Lab 3 Solution