CEP-CCIT
Continuing Education Program
Center for Computing and Information Technology
Fakultas Teknik Universitas Indonesia

Information Search and Analysis Skill (ISAS)

Dynamic Programming Algorithm

Arranged by:

M. Dani Setiawan

M. Naufal Azmi

Novandro Zakaria Pratama

Faculty :

LISTYO PRABOWO ST.

Continuing Education Program Center for Computing and Information Technology

Faculty of Engineering, University of Indonesia

2018

PREFACE

Thank you, the writer wishes to God the Almighty for His blessings and blessings, we can complete this ISAS task both in the form of presentation and paper in a timely manner.

Not forgetting the thanks especially for Listyo faculty and other faculty who always help. Thank you also to fellow students who have supported, and also thank you for being fellow workers in the education at CCIT-FTUI. The ISAS paper entitled ' Dynamic Programming Paint Fence Algorithm' the author submits as a requirement for the ISAS assignment in 2019.

Hope the author, hopefully this paper can be useful for all so that it can add knowledge and insight. The author realizes that this paper is far from perfect. Therefore, the authors really expect all suggestions and criticisms from readers who are constructive in order for the perfection of this paper. Finally, hopefully this paper can provide many benefits for the readers.

Depok, 22 May 2019

Author

TABLE OF CONTENT

TABLE OF FIGURE

CHAPTER I
INTRODUCTION

I.1 Background

Algorithm is an unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing, and automated reasoning tasks. Algorithms can perform calculation, data processing, and automated reasoning tasks. Algorithms often have a loop (iteration) step or require a decision (Boolean logic and comparison) until the task is complete.

Design and analysis of algorithms is a specialized branch in computer science that studies the characteristics and performance of an algorithm in solving problems, regardless of Sorting the Knapsack problem with Dynamic Programming the implementation of the algorithm.

The complexity of an algorithm is a measure of how much computation the algorithm needs to solve the problem. Algorithm that can solve a problem in a short time has a low complexity, while algorithms that take a long time to solve the problem has a high complexity.

In this ISAS, we will explain Dynamic Programming Paint Fence Algorithm. And also, explain how this algorithm works in java application.

I.2 Writing Objectives

The purpose of writing this Paper is to answer the following questions:

1. What is Algorithm?
2. What is Dynamic Programing?
3. What is Paint Fence Algorithm

I.3 Problem Domain

The things that will be discussed in this ISAS:

1. Paint Fence Algorithm Solving

I.4 Writing Methodology

Writing methodology used is to search for sources of information and references from the internet, ask the relatives and read the articles in some resources.

I.5 Writing Framework

Analysis of this ISAS is written with this systematics:

- **CHAPTER I : INTRODUCTION**

  In this section, describe the background, problem analysis, writingobjectives, writing methodology, and systematics of writing.
- **CHAPTER II : BASIC THEORY**

  In this chapter contains theories such as definition, history, basic concepts and related information interms of analysis, especially on problem analysis.
- **CHAPTER III : PROBLEM ANALYSIS**

  This chapter deals with problem analysis such as the definition of shell sort, how shell sort works and the advantages also disadvantages.
- **CHAPTER IV : CONCLUSION AND SUGGESTION**

  In this chapter contains the conclusions of the results of writing and suggestions.
- **BIBLIOGRAPHY**

  In this section will contains the references that we use

CHAPTER II
BASIC THEORY

II.1 Algorithm

The algorithm is a sequence of steps to resolve the problem in a systematic and logical. The algorithm offers a method in solving a problem. The algorithm is defined as a sequence of steps in resolving the problem in a systematic and logical. Approach in a systematic and logical, making the process of problem solving awake his righteousness because the algorithm be correct in order to produce the correct output/solution anyway.

In programming, the important thing to understand is our logic in thinking about how to solve programming problems that will be made. For example, many math problems are easy if completed in writing, but quite difficult if we translate into programming. In this case, the algorithm and programming logic will be very important in solving problems. (Alif, 2015)

II.2 Data Structure

A data structure is proposed to maintain a collection of vertex-disjoint trees under a sequence of two kinds of operations: a *link* operation that combines two trees into one by adding an edge, and a *cut* operation that divides one tree into two by deleting an edge. Each operation requires $O(\log n)$ time. Using this data structure, new fast algorithms are obtained for the following problems:

1) Computing nearest common ancestors.
2) Solving various network flow problems including finding maximum flows, blocking flows, and acyclic flows.

3) Computing certain kinds of constrained minimum spanning trees.
4) Implementing the network simplex algorithm for minimum-cost flow

II.3 Dynamic Programming

Dynamic Programming is a powerful technique that can be usedto solve many problems in time$O(n2)$ or$O(n3)$ for which a naive approach would take exponential time. (Usually to get runningtime below that—if it is possible—one would need to add otherideas as well.) Dynamic Programming is a general approach to solving problems, much like “divide-and-conquer” is a generalmethod, except that unlike divide-and-conquer, the subproblems will typically overlap. This lecturewe will present two ways of thinking about Dynamic Programming as well as a few examples.There are several ways of thinking about the basic idea.

II.4 Dynamic Programming Paint Fence Algorithm

There is a fence algorithm with posts, each post can be painted with one of the colors. The user have to paint all the posts such that no more than two adjacent fence posts have the same color. Return the total number of ways the user can paint the fence algorithm. diff number of combinations with different colors.

Paint Fence is one of the basic Dynamic Programming problems. It can be solve in many ways, and have a variety solution. With this we can learn more about dynamic programming solving, and we can implement it to any code to make it more efficient.

CHAPTER III

PROBLEM ANALYSIS

III. 1 Paint Fence Algorithm

In this problem we can following image depicts the 6 possible ways of painting 3 posts with 2 colors:

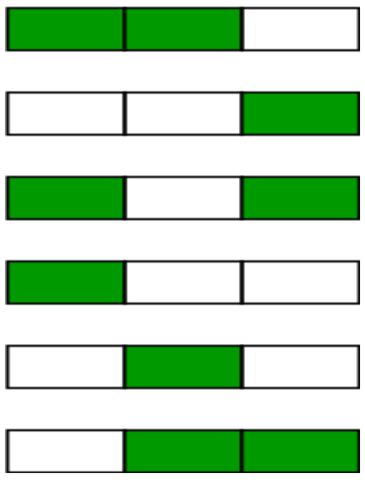
Figure 3.1 Paint Fence

Consider the following image in which c, c' and c" are respective colors of posts i, i-1 and i -2.

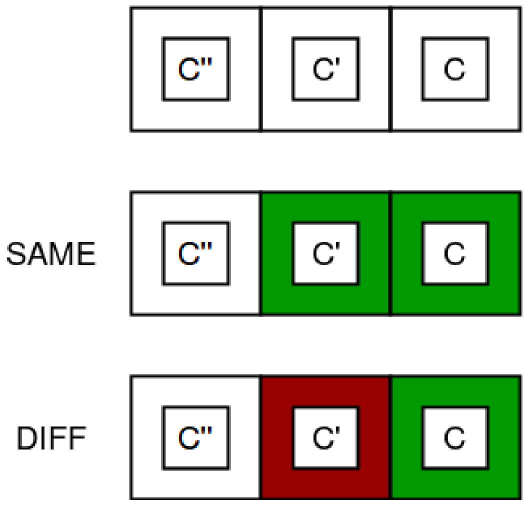

Figure 3.2 Paint Fence

According to the constraint of the problem, c = c’ = c” is not possible simultaneously, so either c’ != c or c” != c or both. There are k – 1 possibilities for c’ != c and k – 1 for c” != c.

```
diff = no of ways when color of last
       two posts is different
same = no of ways when color of last
       two posts is same
total ways = diff + sum

for n = 1
   diff = k, same = 0
   total = k

for n = 2
   diff = k * (k-1) //k choices for
          first post, k-1 for next
   same = k //k choices for common
          color of two posts
   total = k +  k * (k-1)

for n = 3
   diff = [k +  k * (k-1)] * (k-1)
          (k-1) choices for 3rd post
          to not have color of 2nd
          post.
   same = k * (k-1)
          c'' != c, (k-1) choices for it

 Hence we deduce that,
 total[i] = same[i] + diff[i]
 same[i]  = diff[i-1]
 diff[i]  = (diff[i-1] + diff[i-2]) * (k-1)
          = total[i-1] * (k-1)
```

So we can approach this first in the mathematical way . there is a fence(n) and color(k) you need to find how many possibilities there are to paint n fences with k colors. The solution is :

N = 1

there is only k possibilities there for total = k

N = 2

For the same color eac post there is k possibility, so same = k

For the different color we have k*(k-1), so diff =k*(k-1)

And total = k+(k*(k-1))

N = 3

For the same we have, same = k*(k-1)

For different color, diff = (k+(k*(k-1)))*(k-1)

And total = (k*(k-1)) + (k+(k*(k-1)))*(k-1)

And with this we conclude that

Total[i] = same[i] + diff[i]

same[i] = diff[i-1]

diff[i] = diff[i-1] + diff[i-2]*(k-1)

or

diff[i] = total[i-1]*(k-1)

And that we have the counting in mathe matical way. Notice that we can sorten the algorithm with arrays. This is a coomon dynamic programing problems that requires to solve it into little sub problems.

III. 2 Code and Flowchart

This is the code that we use to solve the problems

```
public class Paint_Fence {
    static long countWays(int n, int k)
{
    long total = k;
    int same = 0, diff = k;
    for (int i = 2; i <= n; i++)
    {
        same = diff;
        diff = (int)total * (k - 1);
        total = (same + diff) ;
    }
    return total;
    }
    public static void main(String[] args) {
        // TODO code application logic here
        int n =4;
        int k =3;
        System.out.println(countWays(n,k));
    }
}
```

Output is 66

That is the one variable(total) answer to the solution

We can also put an array to replace the one variable that way we can store the sub problem by adding 1 more to the array

Here is the code :

```
public class Count_try {
    static long Count(int n, int k)
    {
        long dp[] = new long [n+1];
        dp[1]=k;
        int same = 0;
        int diff = k;
        for (int i = 2; i <=n; i++)
        {
            same = diff;
            diff = (int) (dp[i-1]*(k-1));
            dp[i] = same + diff;
        }
        return dp[n];
    }
    public static void main(String[] args) {
        int n =3;
        int k =4;
        System.out.println(Count(n,k));
    }
}
```

- **Flowchart**

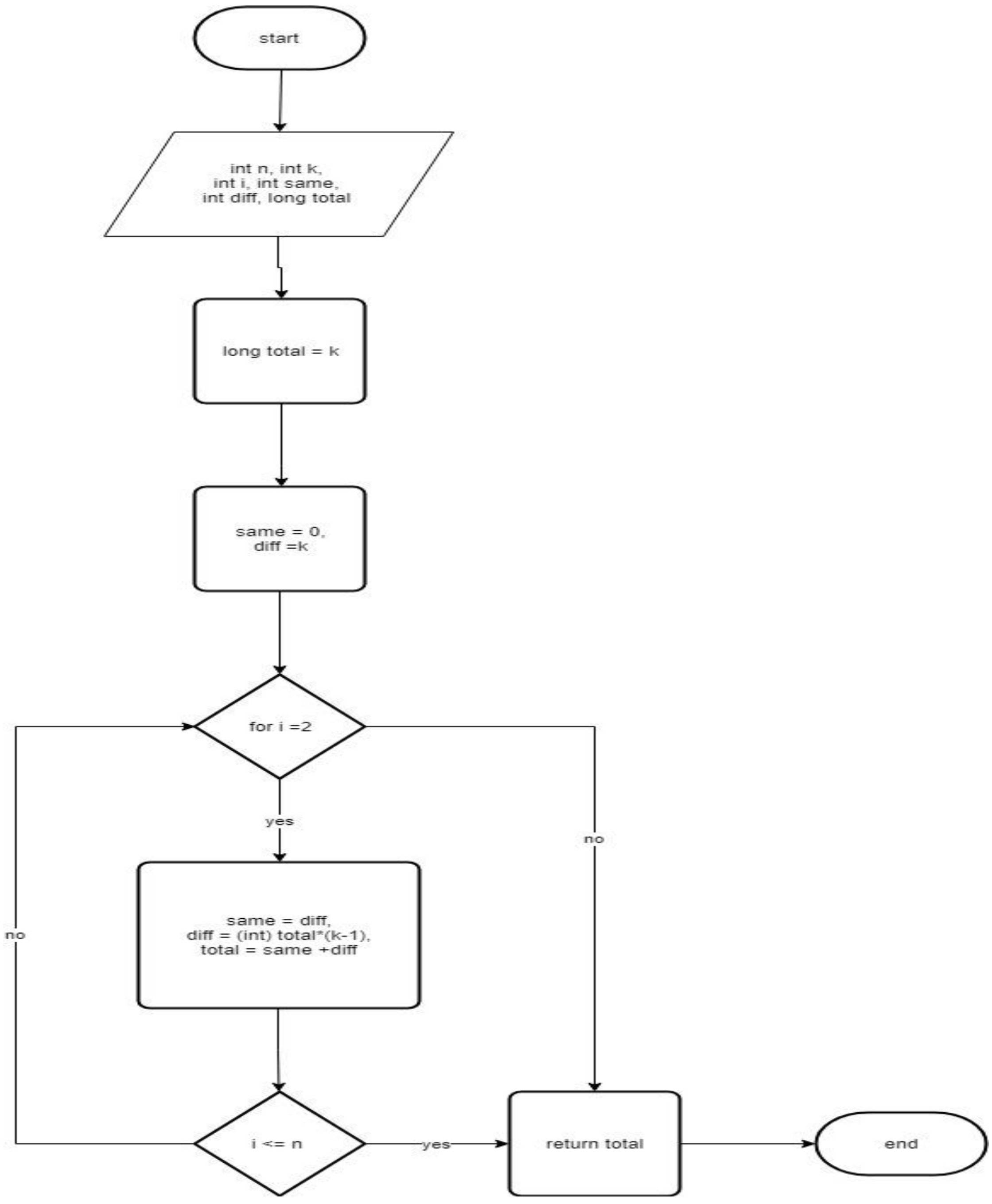

Figure 3.3 Flowchart

Is quite simple and effective way to sole a dynamic programming problem, and that is to break it into subproblem and solve it

CHAPTER IV

CONCLUSION AND SUGGESTION

IV.1 Conclusion

Dynamic programming is a useful technique of solving certain kind of problems. When the solution can be recursively described in terms of partial solutions, we can store these partial solutions and re-use them as necessary (memoization). Running time of dynamic programming algorithm is relatively faster and more efficient.

Paint fence is a simple example of dynamic programming algorithm and problem and we can solve it in quite an easy way. This can be implemented in many ways mostly with divide it to sub problems and store it for more faster run.

IV.2 Suggestion

We suggest for the readers that read this is to apply dynamic programming in their code for more efficient way to solve problems in much faster time consume.

BIBLIOGRAPHY

Alif, M. (2015, November 11). *Jenis jenis sorting*. Retrieved from Techno X Info X Def: http://web.if.unila.ac.id/muhammadalif/2015/11/12/jenis-jenis-sorting/

Abidin, Riswan. (2016, April 25). *Pengertian Algoritma Pemograman*. Retrieved from Tekno Jurnal : https://teknojurnal.com/pengertian-algoritma-pemrograman/

Mishra, Rajat.(2018).*Dynamic Programing*. Retrieved from https://www.geeksforgeeks.org/dynamic-programming/

Lenox, Jerry.(2018).*Paint Fence Algorithm*. Retrieved from https://www.geeksforgeeks.org/painting-fence-algorithm/

University of Nebraska-Lincon.(2010).*Lecture 8 Dynamic Programming.* Retrived from http://cse.unl.edu/~goddard/Courses/CSCE310J/Lectures/Lecture8-DynamicProgramming.pdf

D.Sleator, Daniel & Tarjan,Robert Endre.(1983). *A Data Structure for Dynamic Tress.* Retrieved From https://www.sciencedirect.com/science/article/pii/0022000083900065

Carnegie Mellon's School of Computer Science.(2009).*Lecture 11 Dynamic Programming*. Retrieved from https://www.cs.cmu.edu/afs/cs/academic/class/15451-f10/www/lectures/lect0928.pdf